

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Навчально-науковий Інститут інноваційно-освітніх технологій.

Кафедра комп'ютерних інформаційних технологій.

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Савченко А.С.

“ ” _____ 2020 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИЦІ ОСВІТНЬОГО СТУПЕНЯ “МАГІСТРА”
ЗА СПЕЦІАЛІЗАЦІЄЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА
ТЕХНОЛОГІЇ (ЗА ГАЛУЗЯМИ)”

Тема: «Технологія створення web-сайту інтернет-магазину на базі CMS «Magento».

Виконавиця: Каменяр Світлана Олександрівна.

Керівник: к.т.н., доцент, доцент кафедри КІТ Райчев І.Е.

Нормоконтролер: _____ Райчев І.Е.

Київ – 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Навчально-науковий Інститут інноваційно-освітніх технологій

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології (за галузями)”

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Савченко А.С.

“ ” 2019 р.

ЗАВДАННЯ

на виконання дипломної роботи студентки

Каменяр Світлани Олександрівни

(прізвище, ім'я, по батькові)

1. Тема роботи: «Технологія створення web-сайту інтернет-магазину на базі CMS «Magento»

затверджена наказом ректора № 2701/ст від 22.11.2019 р.

2. Термін виконання роботи: з 25 листопада 2019 р до 20 лютого 2020 р.

3. Вихідні дані до роботи: мова PHP, JavaScript, HTML, CSS, принципи об'єктно-орієнтованого програмування, СКБД MySQL

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): методи та засоби розробки Web-сайту інтернет-магазину на базі системи управління контентом CMS «Magento» з використанням сучасних web-технологій, аналізом та порівнянням із вже існуючими подібними системами

5. Перелік обов'язкового графічного матеріалу: 1. Мета роботи та призначення технології; 2. Аналіз відомих ПП; 3. Структура web-сайту; 4. Реалізація схеми MVC; 5. Логіка відступу; 6. Патерн «Фабрика»; 7. Адміністративна панель

6. Календарний план-графік

<i>№ з/п</i>	<i>Завдання</i>	<i>Термін виконання</i>	<i>Підпис керівника</i>
1.	Аналіз поставлених задач. Формулювання технічного завдання. Виділення основних складових системи.	20.12.19. – 31.12.19	виконано
2.	Аналіз існуючих систем альтернатив, представлених на ринку.	02.01.20 – 06.01.20	виконано
3.	Аналіз інформації кожної складової системи та системи загалом.	08.01.20. – 11.01.20.	виконано
4.	Розробка, розгортання та налаштування системи.	12.01.20. – 22.01.20.	виконано
5.	Оптимізація та додаткове налаштування системи.	23.01.20. – 27.01.20.	виконано
6.	Оформлення пояснювальної записки, підготовка графічних матеріалів	28.01.20. – 08.02.20.	виконано
7.	Підготовка доповіді до захисту дипломної роботи	09.02.20. – 14.02.20.	виконано

7. Консультація з окремого(мих) розділу(ів) роботи:

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв

8. Дата видачі завдання: _____

Керівник _____
(підпис)

Завдання прийняла до виконання _____
(підпис випускниці) (ПІБ)

РЕФЕРАТ

Дипломна робота «Технологія створення web-сайту інтернет-магазину на базі CMS «Magento» викладена на 86 сторінках, містить 25 рисунків, 4 таблиці, 18 наукових джерел.

Об'єкт дослідження – системи управління контентом «Magento».

Метою роботи є дослідження методів та засобів створення Web-сайту для інтернет-магазину з використанням системи управління контентом «Magento», що автоматизує та прискорює перебіг основних бізнес-процесів в цій галузі.

Практична цінність роботи міститься в можливості використання її результатів при розробці та вивченні систем електронної комерції, а саме організації інтернет-магазинів.

Результати дипломної роботи. Досліджувана технологія надала можливості розробки системи e-commerce, яка готова для використання, є достатньо гнучкою для подальшої розробки та удосконалення, містить функції, що досі не були реалізовані в інших програмних продуктах, та реалізована використовуючи сучасні технології, які дозволяють оптимізувати та покращити процес розробки програмного забезпечення в даній галузі.

Перелік ключових слів містить наступні словосполучення:

MAGENTO, CONTENT MANAGEMENT SYSTEM, СИСТЕМА КЕРУВАННЯ КОНТЕНТОМ, ІНТЕРНЕТ-МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, ENTITY-ATTRIBUTE-VALUE MODEL, MODEL-VIEW-CONTROLLER.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ **Ошибка!**

Закладка не определена.

ВСТУП.....	8
Розділ 1. Дослідження систем керування вмістом в галузі e-commerce.....	10
1.1. Система керування веб-контентом.....	10
1.2. Основні функції CMS	12
1.3. Способи роботи	12
1.4. Огляд існуючих рішень	13
1.4.1. VirtueMart.....	13
1.4.2. PrestaShop.....	15
1.4.3. Ubercart.....	18
1.4.4. OpenCart	20
1.4.5. osCommerce.....	21
1.4.6. Magento.....	233
Розділ 2. Розробка проекту веб-сайту на основі складових елементів архітектури Magento.....	26
2.1. Загальний огляд.....	26
2.2. Технологічна база.....	288
2.3. Системні вимоги.....	288
2.4. Огляд основних можливостей адміністративної панелі	299
2.5. Структура сайту магазину.....	30
2.6. Теми дизайну	311
2.7. Пакети та теми в структурі директорій Magento	34
2.7.1. «Base» (базовий) пакет	344

2.7.2. Структура директорій базового пакету.....	355
2.7.3. Інші пакети за замовчуванням	35
2.7.4. Користувацькі теми оформлення.....	36
2.8. Застосування тем	38
2.9. Виключення дизайну	411
2.10. Віджети.....	422
2.11. Блоки.....	444
2.12. Розкладки	445
2.13. Шаблони	500
Розділ 3. Створення веб-сайту інтернет-магазину та власного модулю на базі CMS “Magento”	522
3.1. Встановлення системи на локальний сервер.....	Ошибка! Закладка не определена.2
3.2. Огляд реалізації схеми MVC	533
3.3. Правила створення модулів	55
3.4. Патерн «фабрика»	566
3.5. Події.....	600
3.6. Модель бази даних	622
3.7. Адресація та контролери	667
3.8. Створення модулю	70
3.9. Створення тем.....	744
ВИСНОВКИ.....	777
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	799
ДОДАТОК А	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

БД – база даних

СКБД – Система керування базами даних

SQL – Structured Query Language

CRUD – (англ. Create, Read, Update, Delete) 4 базові функції управління даними «створення, зчитування, зміна і видалення»

ОС – операційна система

ORM – (Object-Relational Mapping) – Об'єктно-реляційна проекція

СКВ (CMS) – (англ. Content Management System) – система керування вмістом

HTML – (HyperText Markup Language) – мова розмітки гіпертекстових документів

PHP – (PHP: Hypertext Preprocessor) – PHP: гіпертекстовий препроцесор

CSS – (Cascading Style Sheets) – каскадні таблиці стилів

EAV – Entity-Attribute-Value – значення атрибута сутності

MVC – (англ. Model-View-Controller) – модель-вид-контролер

ПОП (EDP) – (Event-Driven Programming) – подійно-орієнтоване програмування

URL – (Uniform Resource Locator) – єдиний вказівник на ресурс

XML – (eXtensible Markup Language) – розширювана мова розмітки

WAMP — акронім від «Windows, Apache, MySQL і PHP» — скорочення, що описує комплект пакетів програм під відповідну ОС, позначену першою літерою

ВСТУП

Власник бізнесу, пов'язаного з торгівлею, рано чи пізно стикається з проблемою створення інтернет-магазину. В Інтернеті можна знайти величезну кількість інформації про шляхи вирішення цієї проблеми від власного торгового майданчика на готовому сервісі до повноцінного інтернет-магазину. Вибравши перший варіант, все одно доведеться продовжити пошуки у напрямку справжнього інтернет-магазину. Проблема в тому, що можливості редагування вигляду і функціональності торгового майданчика дуже обмежені, а ваші бізнес-партнери можуть сприйняти такий «інтернет-магазин» не дуже серйозно.

Інтернет-магазин є однією з форм електронної комерції, зазвичай типу B2C – бізнес-до-споживача. B2C включає збирання інформації клієнтами; купівлю фізичних речей чи інформаційних/електронних товарів; і, для інформаційних товарів, одержування товару (програми, гри, електронної книги чи інших творів мистецтва) по електронній мережі. Приклади B2C моделей — мережні компанії продажу в роздріб типу Amazon.com, Drugstore.com, Beyond.com. B2C електронна комерція зменшує ціну угод (особливо ціну пошуку), збільшуючи доступ споживачів до інформації й дозволяючи споживачам знайти найбільш конкурентоспроможну ціну за товар або послугу. B2C електронна комерція також зменшує ринкові бар'єри входу, тому що вартість створення й розкручування сайту набагато менша ніж установка структури фірми. У випадку інформаційних товарів, B2C електронна комерція ще привабливіша, тому що це зберігає фірми від фактору додаткової вартості фізичної мережі розподілу. Крім того, для країн із зростаючою кількістю користувачів Internet, поставка інформаційних товарів стає усе більш і більш доступною.

Протягом останніх 5-ти років український ринок електронної комерції показував щорічне зростання на рівні 50% незалежно від перманентних економічних коливань. При цьому ринок ecommerce володіє серйозним потенціалом. Дійсно, у Німеччині ємність ринку електронної комерції становить

близько \$35млрд. В Україні аналогічний показник становить не більше \$500-\$600млн.

Основними факторами, які гальмують розвиток українського ринку електронної комерції, є такі:

- недостатність законодавчої бази, щоб врегулювати процеси купівлі/продажу онлайн, здійснювати електронні платежі за оплачені товари/послуги та встановлювати єдині прозорі правила гри на ринку (як для продавців, так і для покупців);
- слабо розвинена національна система електронних платежів;
- досить низька ефективність більшості існуючих українських торговельних майданчиків;
- недостатній рівень проникнення інтернету (особливо в регіонах).

Веб-сайти часто створюють на базі систем керування вмістом. Такий підхід дозволяє значно скоротити час розробки, тому що більшість потрібного функціоналу уже реалізована.

Система керування вмістом – це інструментарій призначений для зручного наповнення і подальшого управління вмістом (контентом) сайту. Як правило, усі CMS включають у себе можливість керування шаблоном сайту, тобто тим, як в кінцевому результаті виглядатиме ваш сайт. Для популярних CMS існують множини готових шаблонів і крім того, є можливість створювати власні, індивідуальні шаблони сайтів.

Розділ 1. Дослідження систем керування вмістом в галузі e-commerce

Система керування вмістом (СКВ; Content Management System, CMS) — це програмне забезпечення для організації веб-сайтів або інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

CMS значно спрощують створення та підтримку сайтів.

1.1. Система керування веб контентом

Система керування веб контентом (Web Content Management System) – це програмне забезпечення яке надає адміністративні інструменти для організації авторизації, співпраці та інші, що дозволяють користувачам з незначними знаннями в програмуванні створювати та керувати контентом сайту з відносною простотою.

Шар представлення (Presentation Layer (Template Engine)) відображає контент сайту відвідувачам на основі набору шаблонів.

Більшість систем використовує кешування на стороні сервера для покращення швидкодії.

Адміністрування також зазвичай проводиться через веб-інтерфейс, але деякі системи потребують використання «товстого» клієнту.

Подібні системи дозволяють технічно не підготовленим користувачам робити зміни на сайті після проходження тренувального курсу. Для налаштування та додавання нових функцій на сайті зазвичай потрібен системний адміністратор чи веб-розробник.

Існують сотні доступних CMS. Завдяки їхній функціональності, CMS можна використовувати в різних компаніях. Незважаючи на широкий вибір інструментальних та технічних засобів, наявних в CMS, існують загальні для більшості типів систем характеристики.

Кафедра КІТ (47)				НАУ 20 02 87 .000 ПЗ				
Виконала	Каменяр С.О.			ДОСЛІДЖЕННЯ СИСТЕМ КЕРУВАННЯ ВМІСТОМ В ГАЛУЗІ E-COMMERCE	Літера	Арк.	Аркушів	
Керівник	Райчев І.Е.						10	16
Консульт.								
Н.котрол.	Райчев І.Е.							
.	.							
					УС-201Мз 122			

Більшість сучасних CMS мають модульну архітектуру, що дозволяє адміністратору обирати та налаштовувати ті компоненти, які йому необхідні. Типові модулі: динамічне меню, блог, новини, опитування, пошук по сайту, статистика відвідувань, гостьова книга та інші.

Сайти, організовані засобами систем управління контентом, засновані на наступних технологіях: веб-сервер, сховище даних (часто СУБД, наприклад MySQL чи PostgreSQL, але існують і noSQL-CMS), веб-застосунок (веб-додаток) для забезпечення роботи системи, візуальний (WYSIWYG – what you see is what you get – що бачиш те й отримуєш) редактор сторінок, файловий менеджер з веб-інтерфейсом для управління файлами сайту, система управління правами користувачів та редакторів сайту.

Найбільш поширені наступні технологічні платформи які використовуються в якості основи веб-додатку, який реалізує роботу CMS: PHP, Perl, .NET.

Багато сучасних СКВ поширюються як безкоштовні і легкі у встановленні (інсталяції) програми, які розробляються групами ентузіастів під вільними ліцензіями такими як GNU/GPL, MIT чи Apache. Є багато компаній які надають послуги створення та підтримки сайтів на базі таких систем.

Існує термін контент-менеджер, який відображає вид професійної діяльності, тобто співробітника який відповідальний за наповнення (контент) сайту. CMS дозволяють контент-менеджеру не мати навичок програмування.

Більшість сучасних систем керування контентом мають візуальний редактор, який дозволяє користувачам створювати та редагувати сторінки використовуючи спрощену розмітку.

Перші СКВ були розроблені у великих корпораціях для організації роботи з документацією. У 1995-му від компанії CNET відокремилася окрема компанія Vignette, яка започаткувала ринок для комерційних СКВ. З часом діапазон продукції розширювався і все більше інтегрувався у сучасні мережеві рішення аж до популярних веб-порталів.

1.2. Основні функції CMS

- Надання можливості управління сайтом з мінімальним рівнем підготовки, можливо навіть створення сайту з уже готових блоків.
- Додавання, редагування, видалення інформації з сайту.
- Організація спільної роботи з контентом.
- Керування режимом доступу користувачів.
- Формування веб-сторінок.

1.3. Способи роботи

- *Генерація сторінок за запитом.* Системи такого типу працюють на основі зв'язки «модуль редагування → база даних → модуль представлення». Модуль представлення генерує сторінку з контентом при запиті на нього на основі інформації з бази даних. Інформація в БД змінюється за допомогою модуля редагування. Сторінки заново створюються сервером при кожному запиті, а це створює навантаження на сервер. Але це навантаження може бути багатократно зменшене застосуванням різних методів кешування.

- *Генерація сторінок при редагуванні.* Системи цього типу при редагуванні сторінок вносять зміни у вміст сайту та створюють набір статичних сторінок. При такому способі втрачається інтерактивність між відвідувачами сайтів та контентом даного сайту.

- *Змішаний тип.* Як зрозуміло із назви, цей тип поєднує в собі переваги перших двох. Може бути реалізований шляхом кешування — модуль представлення генерує сторінку один раз, надалі вона через деякий час буде в декілька разів швидше завантажуватися із кешу. Кеш може оновлюватись як автоматично, через деякий час чи при внесенні змін у певні розділи сайту, так і вручну за командою адміністратора. Другий підхід — збереження певних інформаційних блоків на етапі редагування сайту і збирання сторінок з цих блоків при запиті відповідної сторінки користувачем.

1.4. Огляд існуючих рішень

Розглянемо деякі з популярних систем управління контентом які використовуються в галузі електронної комерції.

1.4.1. VirtueMart

VirtueMart, раніше відомий як mambo-phpShop, вільне програмне забезпечення для створення Інтернет-магазину, створене для доповнення таких систем керування вмістом веб-сайту, як Mambo та Joomla. VirtueMart написаний на PHP і використовує базу даних MySQL. Найбільш підходить для веб-сайтів з низьким або середнім рівнем завантаження. VirtueMart захищений ліцензією GNU GPL. VirtueMart з'явився вперше у вигляді автономної програми phpShop для створення інтернет-магазину. Після того як Mambo відгалузилося в Joomla, розробник поміняв бренд mambo-phpShop на VirtueMart, який офіційно підтримує більш нову систему управління сайтами Joomla. Поточні версії VirtueMart можуть функціонувати з Mambo.

VirtueMart підтримує необмежену кількість продукції і категорій, необмежену кількість валют для одного продукту, призначення продукції безлічі категорій, можливість продавати продукцію, призначену для скачування, а також надає можливість відключити функцію продажу і використовувати VirtueMart в режимі каталогу. VirtueMart дозволяє мати різні ціни для однієї продукції, засновані на кількості чи приналежності покупця до певної групи, і надає можливість використання різних платіжних систем.

У VirtueMart реалізована система знижок, купонів, а також великий вибір систем оплати та способів доставки.

При перенесенні VirtueMart з Mambo на Joomla розробники не приділили належної уваги відмінностям між Mambo та Joomla, в результаті VirtueMart гілок 1.0.x містить у собі багато старого процедурного коду і компоненти, вже існуючі в Joomla, але з якихось причин не використовувані (шаблонізатор, з'єднання з базою і т. д.).

Панель адміністратора:

- Маркетингові інструменти
 - Багаторівневе ціноутворення
 - Купони
 - Правила ціноутворення
- Визначення податкових ставок
 - Визначення правил для регіону чи країни.
 - Створення класів податку, наприклад «Нормальний» або «Оптовий»
 - Виставлення ставки ПДВ.
- Продукти та каталог
 - Можливість завдання різних атрибутів (властивостей) для товару.
 - Необмежена вкладеність категорій.
 - Конфігуровані продукти - продукти з вибілковими властивостями (наприклад: колір, розмір тощо. Д.)
 - Сортуння товару по наперед визначеним атрибутам (виконується за допомогою додаткового модуля).
- Адміністрування
 - Групи покупців (прості, VIP і т. Д.)
 - Імпорт / Експорт (у вигляді сторонніх розробок)
 - Система контролю доступу
 - Вибір валюти
 - Вибір країн
 - Звіти продажів
 - Вибір способів доставки
 - Вибір способів оплати

Можливості фронт-енду:

- Пропозиція покупцям супутніх товарів
- Пропозиція покупцям недавно переглянутих товарів
- Пошукова оптимізація (SEO) - поки у вигляді сторонніх розробок
- Зв'язок з покупцем
 - Повідомлення по електронній пошті
 - RSS-підписка на категорію або всі групи товарів
- Пошук і розширений пошук
- Порівняння продуктів - у вигляді сторонніх розробок
- Теги для продуктів - у вигляді сторонніх розробок
- Відгуки покупців про товари
- Оформлення замовлення
- Кошик покупок
- Акаунти покупців

- Рекомендація товару одним покупцем іншому
- Топ-10 товарів
- Відображення спеціальних пропозицій товару

Імпорт та експорт даних:

У стандартну збірку не входять модулі імпорту / експорту, тільки у вигляді сторонніх розробок.

- Імпорт та експорт товарів
- Імпорт та експорт товарних залишків на складах
- Імпорт та експорт покупців

Можливість створення шаблонів імпорту/експорту для різних цілей. Існують сторонні розробки модулів експорту в Яндекс.Маркет та інші.

Способи оплати:

- Кредитною картою
- Оплата при доставці
- PayPal
- eCheck.net
- MerchantWarrior
- Сторонні розробки модулів оплати, серед яких є WebMoney, Яндекс.Деньги, RBKMoney, Qiwi Visa Wallet, PayKeeper, Assist та інші.

Способи доставки:

- auspost
- Канада Повідомлення
- DHL
- FedExdc
- Flex
- InterShipper
- shipvalue
- Міжнародний поштовий союз
- USPS
- Стандартний модуль доставки
- Доставка в залежності від регіону чи країни
- Самовивіз

1.4.2. PrestaShop

PrestaShop – веб-додаток для електронної комерції з відкритим кодом.

Написаний на PHP, для написання шаблонів використовується Smarty, розрахований на MySQL. Цей двигун призначений для малого та середнього бізнесу та має більше 310 стандартних функцій для швидкого створення функціонального магазину. В 2010-2014рр. визнаний кращим веб-додатком для інтернет-магазинів, завоювавши нагороду Best Open-source Business Application.

Перекладений майже на 60 мов, спільнота нараховує більше 500.000 чоловік.

Існує платна і безплатна підтримка. Штаб квартири компанії знаходяться в Парижі та Майямі. Команда налічує більше 100 чоловік та складається із розробників, дизайнерів та ІТ-спеціалістів в електронній комерції.

PrestaShop інколи порівнюють з Magento та OpenCart. Поступається першому розміром спільноти та кількістю готових рішень, але при цьому легший в освоєнні. OpenCart як і PrestaShop легкий в освоєнні, та набирає популярність з кожним роком.

Функціональність

- Каталог
- Оплата
- Доставка товару
- Статистика
- Переклади
- Локалізація
- Безпека
- Управління
 - Текстовий редактор WYSIWYG
 - Можливість додавання додаткових модулів
 - Резервне копіювання бази даних (часткове або повне)
 - Автоматична генерація htaccess -файлів
 - Автоматична генерація файлів robots.txt
 - Відправлення електронної пошти через SMTP (підтримується SSL і TLS) або за допомогою функції PHP mail.
 - Управління шрифтами в PDF -файлах
 - Індексція товарів для оптимізації пошуку
 - Управління запасами

- СМС (інвентаризація, новий порядок ...)
- Немає в наявності сповіщень електронною поштою
- Модулі магазину
 - Аякс -коризина
 - Хмара міток
 - Аякс Пошук
 - Псевдонім -пошук
 - Настроюються сторінки (текст, фото)
 - Продукти сцени (презентація продукції)
 - Поради по продуктам сторінок: дата придбання або додавання в корзину
 - Модулі On-Off
 - Товари, що рекомендуються на головній сторінці
 - Можливість вставки оголошень
 - Показати нові товари
 - Можливість інтеграції з Google Adsense
 - Закладка в один клік
 - Вибір валюти
 - Вибір мови
 - Продукти RSS -канал
 - Показати інші товари з тієї ж категорії
 - Показати кращих продавців
 - Відображення продуктів категорії
 - Відображення додаткового посилання
 - Показати виробників
 - Показати постачальників
 - Блок «Мій рахунок»
 - Дисплей рахунки повного клієнта (інформація, замовлення, ковзає ...)
 - Підписка на розсилку новин блок
 - RSS-канал для підписки
 - Підкреслюючи доступних платіжних засобів
 - Третій блок RSS канал
 - Швидкий пошук

1.4.3. Ubercart

Ubercart є вільним програмним забезпеченням для відкриття торгових майданчиків, інтернет-магазинів, аукціонів, створюваних для доповнення такої системи керування вмістом, як CMF Drupal. Модуль Ubercart може бути встановлений на будь-якому сервері з підтримкою PHP і MySQL і поширюється по ліцензії GNU GPL.

Ubercart повністю інтегрований з Drupal'ом, що означає 100% сумісність з ним будь-якого веб-сайту, що використовує Drupal. Модуль може використовуватися як у випадках продажу товарів і послуг, так і при завантаженні будь-якої онлайн продукції. Крім того, його можна застосовувати для найму та працевлаштування, створення значущих подій. Головним же призначенням Ubercart є можливість відкривати необмежену кількість платіжних шлюзів різних електронних світових валют і банківських систем. Ubercart дозволяє проводити цілий ряд комплексних дій всередині CMF Drupal, а саме: додавати новий продукт, здійснювати мультипродаж товарів, робити знижки і видавати купони, обмінювати валюту, здійснювати будь-які внутрішні і міжнародні платежі, підтримувати партнерські програми, виплачувати комісії, видавати купони на ліцензовані товари та послуги, проводити комплексну класифікацію продукції підприємства.

Також як і Drupal, Ubercart може працювати на різних мовних платформах.

Drupal — популярна вільна модульна система керування вмістом (СКВ) з відкритим сирцевим кодом, написана на мові програмування PHP.

Завдання систем керування вмістом — полегшити створення, наповнення та оновлення веб-сайту. Drupal може працювати на таких популярних системах як Windows, Mac OS X, Linux, власне, на будь-якій платформі, яка підтримує роботу веб-сервера Apache, Nginx, Lighttpd або Microsoft IIS; також потрібна наявність системи керування базами даних MySQL/MariaDB, PostgreSQL, SQLite чи інших комерційних БД. Повні системні вимоги Drupal наведені на офіційному сайті.

Критики Drupal ставлять в докір розробникам слабе використання об'єктних можливостей PHP. API Drupal практично не використовує наявні в PHP можливостей ООП. Розробники аргументують це слабкою реалізацією ООП у мові (особливо до версії PHP 5). Об'єктна модель в Drupal присутня, але в дещо нетрадиційному для PHP вигляді.

До недоліків (але водночас і до переваг) Drupal можна віднести відсутність зворотної сумісності API при досить високій динаміці розробки проекту. Практично в кожному релізі відбуваються зміни API, коли поряд з додаванням нових функцій прибираються деякі старі або змінюються параметри виклику функцій. Це призводить до необхідності розробникам сторонніх модулів адаптувати їх для роботи з новими версіями Drupal. Проте зміни API і процедура адаптації модулів до нових версій описуються в документації для кожного релізу, а також завжди пропонується механізм автоматизованого апгрейда ядра системи на нову версію. Плюс даної схеми розробки — немає необхідності тягти з версії у версію програмний шар сумісності зі старими API, що полегшує поточний код системи.

У дистрибутив системи входить набір модулів, що дають наступні можливості:

- збір інформаційних стрічок (RSS, RDF, Atom);
- ведення блогів, підшивань і форумів;
- створення форм для відправки повідомлень;
- локалізація системи;
- перейменування посилань (призначення посиланням зрозумілих і зручних псевдонімів);
- проведення опитувань;
- призначені для користувача профілі, що настроюються;
- пошук за змістом (за зміст вважається і повідомлення на форумах, і сторінки, і будь-які інші призначені елементи);
- ведення журналу статистики (відвідуваності);
- таксономія (впорядковування матеріалу за категоріями) — дуже «цінна» можливість;
- включення механізму авторегулятора контролю навантаження та інші.

1.4.4. OpenCart

OpenCart — це система інтернет-торгівлі з відкритим кодом, яка розповсюджується за ліцензією GNU General Public License.

OpenCart може бути встановлено на будь який веб-сервер Apache з підтримкою PHP5 і вище та MySQL. Навколо OpenCart сформована велика спільнота (понад 46.000 учасників), завдяки якій створено понад 8.500 безкоштовних розширень у вигляді додаткових модулів.

Найвагомішими перевагами OpenCart над системами Magento, Virtuemart і osCommerce є сучасна MVC-архітектура, підвищена швидкість роботи, vQmod, багатофункціональна адміністративна панель управління контентом, та менше споживання серверних ресурсів.

OpenCart добре зарекомендував себе в комерційному секторі як надійна і недорога в обслуговуванні система електронної торгівлі, що має підтримку розрахунку всіх найвідоміших систем електронної оплати.

Основні можливості.

Технічні переваги:

- Підтримка PHP 5.x і MySQL 4.x, 5.x.
- Код відповідає основним принципам шаблону Model-View-Controller, який дозволяє проводити роботи різної складності незалежно одна від одної.
- Порівняно з конкурентами (Magento, Virtuemart, osCommerce) має кращі показники швидкості і несе менше навантаження на сервер.
- Підтримка багатьох сучасних браузерів.
- Вбудована багатомовність. Доступна українська мова.
- Необмежена кількість категорій і товарів.
- Підтримка шаблонів, модулів і доповнень.

Адміністрування/База:

- Підтримка одного і більше магазинів.
- Необмежена кількість продуктів і категорій.
- Підтримка фізичних і віртуальних (з можливістю завантаження) товарів.
- Легкість резервного копіювання і відновлення бази даних.
- Статистика товарів, замовлень і продажів.
- Багатомовність.
- Підтримка валют і курсів.

Клієнтська частина:

- Реєстрація покупців.
- Усі замовлення зберігаються в базі даних для ефективного пошуку історії покупок.
- Клієнти можуть переглядати історію і статус своїх замовлень.
- Тимчасовий кошик для гостей і постійний для зареєстрованих клієнтів.
- Швидкий і зручний інструмент пошуку.
- Підтримка SSL (Secure Sockets Layer).
- Зручна навігація по сайту.
- Клієнт може мати декілька адрес доставки в персональній адресній книзі.

Система оплати і доставки:

- Підтримка багатьох типів платежів (чеки, платіжні доручення).
- Підтримка багатьох платіжних систем за допомогою модулів (2Checkout, PayPal, Authorize.Net, iPayment, RuPay, Webmoney, Приват24).
- Налаштування методів оплати для різних регіонів.
- Розрахунок доставки на базі ваги і ціни товару та зони доставки.
- Безліч модулів розрахунку доставки.
- Розрахунок податків з урахуванням регіону.

Варто звернути увагу:

- Шаблонізація на базі PHP.
- У цей час не оптимізована робота SEO.
- Швидкі темпи розвитку версій, що потребує частого поновлення ядра.

1.4.5. osCommerce

osCommerce («Open Source Commerce») — це движок інтернет-магазину. Він може бути встановлений на будь-якому веб-сервері з підтримкою PHP і MySQL. Це вільне програмне забезпечення з GNU General Public License.

Навколо osCommerce сформувалося величезне співтовариство (більш 140,000 учасників) , завдяки якому існує більше 4000 контрибуцій (різних модулів для osCommerce), що дозволяють змінювати і доповнювати функції магазину самим різним чином. По всьому світу функціонують десятки тисяч магазинів на базі osCommerce (14,063 магазину офіційно зареєстровані в каталозі).

Основні можливості:

- Сумісно з PHP 4.x, 5.x і MySQL 4.x, 5.x.
- Сумісність з усіма основними браузерами.
- Вбудована багатомовність, за замовчуванням встановлені англійський, німецький, іспанська мови. Доступні російська, українська і багато інших.
- Майстер інсталяція (майстер).
- Необмежене число розділів і товарів.

Administration/Base:

- Підтримує необмежену кількість продуктів і розділів категорій.
- Підтримка фізичних і віртуальних (завантажуються) товарів.
- Легкість резервного копіювання і відновлення даних.
- Статистика товарів і замовників.
- Багатомовна підтримка.
- Підтримка декількох валют.

Клієнтська частина:

- Реєстрація покупців.
- Всі замовлення зберігаються в базі даних для швидкого й ефективного пошуку (історія покупок для покупців).
- Клієнти можуть переглядати історію і статуси своїх замовлень.
- Тимчасова корзина для гостей і постійна для клієнтів.
- Швидкий і дружній інтерфейс пошуку.
- Безпека з підтримкою SSL (Secure Sockets Layer).
- Клієнт може мати кілька адрес доставки в своїй адресній книзі.

Система оплати і доставки:

- Підтримка численних типів платежів (чеки, платіжні доручення).
- Підтримка численних платіжних систем (модулям) (2Checkout, PayPal, Authorize.Net, iPayment, RuPay, Webmoney).
- Налаштування методів оплати для різних областей.
- Розрахунок доставки на основі ваги та ціни товару, зони доставки.
- Розрахунок податків.

На жаль, osCommerce не володіє системою шаблонів (до версії 3 Alpha 5) для настройки дизайну, але є ряд модулів які вирішують дану задачу. Наприклад, STS і BTS. Також система шаблонізатор є у багатьох клонів osCommerce.

1.4.6. Magento

“Magento” є професійною системою для управління інтернет-магазинами. Згідно даних Alexa, Magento — це найпопулярніша система управління інтернет-магазинами в світі на лютий 2019. Ще у 2012р. компанія Magento Inc. була придбана компанією eBay Inc.

Зовнішній вигляд вітрини в Magento визначається так званими *темами*:

- набір шаблонів (templates) відображення видимих на екрані блоків;
- набір правил, що визначають, які блоки в якому місці конкретної сторінки відображати, і об'єднаних у файли логічної розмітки (layouts);
- набір ресурсних файлів (скінів): CSS, картинок, скрипти на мові JavaScript.

На відміну від більшості інших систем управління сайтами (наприклад, Joomla), в Magento теми можуть бути пов'язані кожна з одним відношенням успадкування: якщо в поточній темі не визначений конкретний шаблон, правило відображення або ресурсний файл, то система бере їх з базової теми.

Так, наприклад, всі три стандартних теми для **Magento**: Default, Modern і iPhone успадковуються від єдиної базової теми. Сторонній програміст може створити свою базову тему.

На відміну від Joomla, де прямо в адміністративній частині можна призначити місце відображення і параметри конкретного модуля на сторінці, Magento в основному націлена на редагування екранних блоків вітрини *не* через адміністративну частину, а через *файли логічної розмітки і файли шаблонів*. Такий підхід, з одного боку, дає велику гнучкість розробникам магазинів, але, з іншого боку в цьому одна з причин, чому у простих адміністраторів Magento вважається складною у вивченні системою.

Можливості

Основні:

- З єдиної адміністративної частини можна управляти торговою мережею інтернет-магазинів на різних доменах, мовах, з різним товарним асортиментом.
- Гнучка настройка зовнішнього вигляду («теми» і «шаблонів») вітрини.
- Відображення цін на вітрині в різній валюті (у тому числі за вибором покупця).

- Багатомовність.
- Методи оплати і доставки.

Панель адміністратора:

- Маркетингові інструменти.
- Багаторівневе ціноутворення.
- Купони.
- Можливість організації розпродажів, налаштування оптових знижок
- Звіти з продажів, по корзинах покупця та про список відзначеного товару.
- Звіт про відкликання.
- Звіт за тегами та по Пошуку.
- Визначення податкових ставок.
- Визначення правил для регіону, країни або поштового коду.
- Створення класів податку, наприклад «Нормальний» або «Оптовий».
- Визначення правил податку, наприклад «Податок на одяг», також можна пов'язати різні групи товарів з класами податків.
- Продукти та каталог.
- Можливість завдання різних атрибутів (властивостей) для товару.
- Угрупування товару в комплекти.
- Конфігуровані продукти - продукти з вибірковыми властивостями (наприклад колір, розмір, тощо).
- Сортування товару по наперед визначеним атрибутам. Атрибути доступні для сортування визначаються в панелі адміністрування.
- Система управління контентом.
- Групи покупців.
- Імпорт / Експорт (В даний момент імпорту товарів працює в обмеженому варіанті, можливе внесення товарів, але не оновлення).
- Система контролю доступу.

Валюти

- Можна вести облік (складський і управлінський) в одній валюті, а ціни на вітрині показувати в іншій.
- Можна призначити різним вітринам різні валюти цінників.
- Можна дати відвідувачам можливість перемикатися між валютами.
- Можна налаштувати оновлення валютних курсів за розкладом.
- Налаштування відображення валют визначаються поточною локалізацією (локаллю) відвідувача (використовується бібліотека Zend Locale).

Можливості пошукової оптимізації

Стандартна збірка Magento:

- Дає повний контроль над розміткою HTML. Є можливість установки індивідуального шаблону HTML для конкретного товару або товарного розділу.
- Дозволяє для кожної сторінки (товару, розділу або текстової) задавати індивідуальні мета-теги description і keywords.
- Дозволяє для кожного товару і товарного розділу задавати індивідуальне і настраювати закінчення адреси (URL Key).
- Дозволяє управляти для кожної сторінки заголовком TITLE (мається як автоматичний режим, так і ручний).
- Автоматично створює для сайта карту в форматі XML для пошукових ботів.

Пошукова оптимізація товарних зображень. Magento при створенні сторінок вітрини описує товарні зображення атрибутом ALT тега IMG . Адміністратор для кожного товарного зображення може або вручну вказати опис, або дозволити Magento описати товарне зображення автоматично, вико

Розділ 2. Розробка проекту веб-сайту на основі складових елементів архітектури “Magento”

2.1. Загальний огляд

Magento — платформа електронної комерції для створення інтернет-магазинів з відкритим кодом, яка поширюється згідно з OSL 3.0.

«Magento» — це одна з найпопулярніших відкритих систем для організації електронної комерції в мережі: на базі цієї платформи створено понад 150 тисяч інтернет-магазинів, сторонніми розробниками створено більше 2500 розширень, спільнота проекту налічує близько 400 тисяч учасників, код платформи був завантажений більше 3,5 млн. разів, через системи на базі Magento кожного року продається товарів більш як на 28 мільярдів доларів [1]. За кілька років свого існування платформа кілька разів була лауреатом премій «Best of Open Source Software Awards» і «SourceForge Community Choice Awards» [2].

Програмне забезпечення «Magento» створене з використанням Zend Framework. Сирцевий вихідний код Magento поширюються в рамках схваленої організацією Open Source Initiative (OSI) ліцензії Open Software License (OSL) v3.0, близької за своєю суттю до ліцензії AGPL, але не сумісної з GPL. Особливість ліцензії OSL полягає в тому, що якщо хтось модифікує програму, але не буде поширювати результат своєї роботи і просто задіює його на своєму сайті, то він все одно зобов'язаний відкрити код внесених змін і поширювати його під тією ж ліцензією. При цьому ліцензія OSL не забороняє комерційне використання програмних продуктів.

Діаграма варіантів використання інтернет-магазину, яка розроблена в дипломній роботі, наведена на рисунку 2.1, а відповідна діаграма класів на рисунку 2.2. Система керування контентом «Magento», значно спрощує процес створення інтернет магазину будь-якого профілю.

Кафедра КІТ (47)				НАУ 20 02 87. 000 ПЗ			
Виконала	Каменяр С.О.			РОЗРОБКА ПРОЕКТУ ВЕБ-САЙТУ НА ОСНОВІ СКЛАДОВИХ ЕЛЕМЕНТІВ АРХІТЕКТУРИ “MAGENTO”	Літера	Арк.	Аркушів
Керівник	Райчев І.Е.					26	26
Консульт.					УС-201Мз 122		
Н.котрол.	Райчев І.Е.						
.	.						

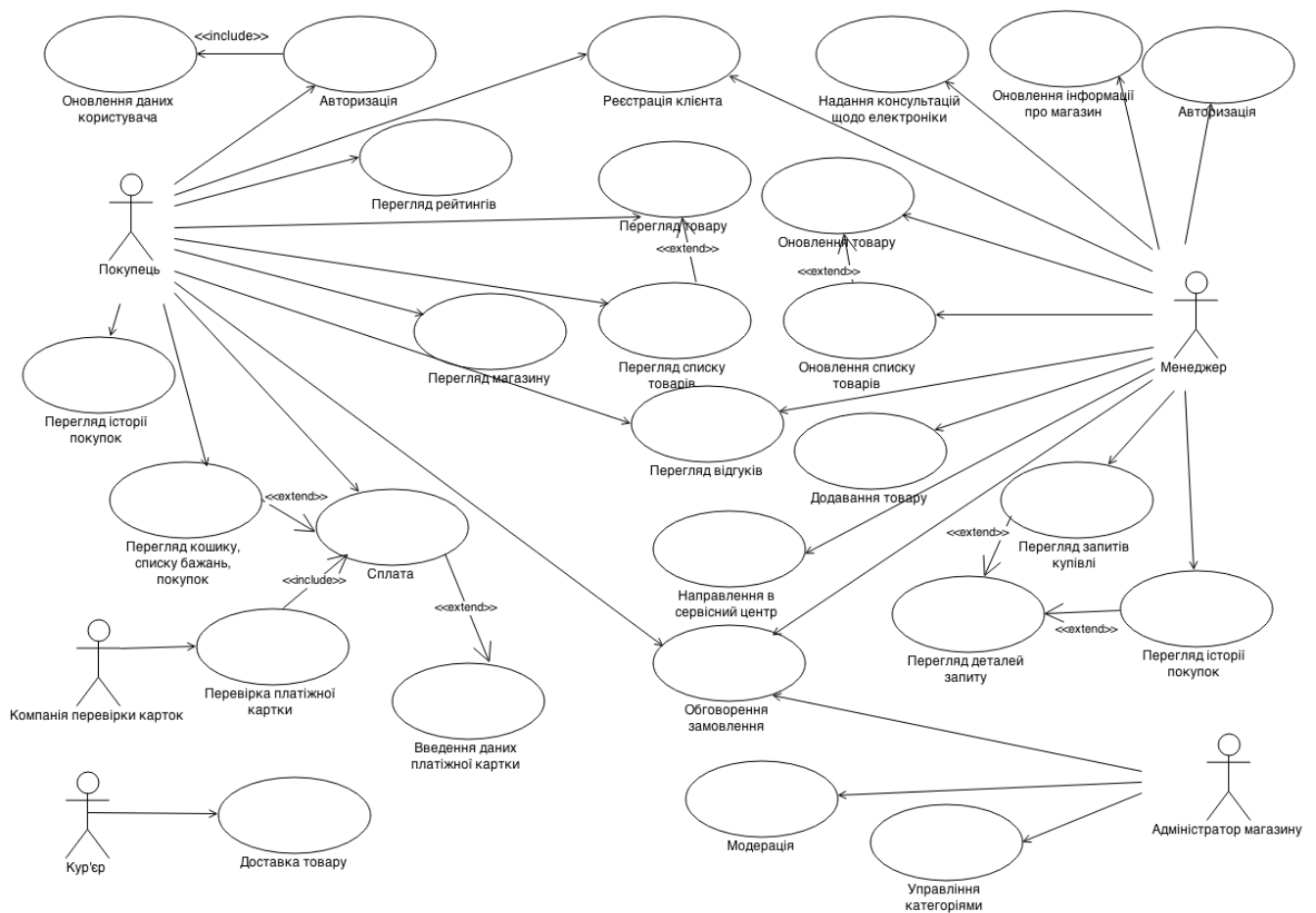


Рис. 2.1. Діаграма варіантів використання інтернет-магазину



Рис. 2.2. Діаграма класів інтернет-магазину

Відомо, що декілька великих українських інтернет-магазинів використовують Magento, серед них:

- Comfy [3] (comfy.ua, виручка 4,2 млрд. грн за перше півріччя 2019 року [4]).
- Allo [5] (обіг в 2018 році — \$29,5 млн. [6]).
- Мобілочка [7] та інші.

2.2. Технологічна база

Як і будь-який інший веб-додаток, Magento має клієнт-серверну архітектуру. Backend, тобто серверна частина написана на PHP та SQL. Причому в якості системи управління базами даних може використовуватись не тільки MySQL як у більшості конкурентів (та й у ранніх версіях Magento).

PHP (Hypertext Preprocessor — гіпертекстовий препроцесор) на даний момент є найпоширенішою мовою створення веб-додатків та підтримується переважною більшістю хостинг-провайдерів, що є безсумнівним плюсом. MySQL також підтримується більшістю хостинг-провайдерів тому проблем з впровадженням системи виникнути не повинно.

Для Frontend, тобто клієнтської частини у Web, стандартом де-факто є стек HTML, CSS, ECMAScript (JavaScript).

За рахунок того що PHP є високорівневою мовою програмування з великою бібліотекою реалізованих функцій, при розробці можна зосередитись на програмуванні бізнес-логіки, а не витратити час на різні низько-рівневі речі.

2.3. Системні вимоги

- Linux або інша UNIX-сумісна операційна система (Windows також підтримується, але спочатку варто ознайомитися з усіма відомими проблемами).
- Веб-сервер Apache 2.x або Nginx 1.7.x.
- PHP 5.4 та вище.
- MySQL 5.6 та вище також підтримується Oracle або Percona.

- Sendmail-сумісний поштовий сервер. Magento може з'єднатися безпосередньо з сервером по протоколу SMTP.

Цим вимогам відповідає більшість сучасних хостингів.

2.4. Огляд основних можливостей адміністративної панелі

Зручність та набір можливостей адміністративної панелі є важливою частиною системи інтернет-магазину на базі Magento [8-11]. На головній сторінці адміністративної панелі відображені деякі зведені дані діяльності магазину, такі як топ пошукових запитів та найбільш продавані товари (рис. 2.3).

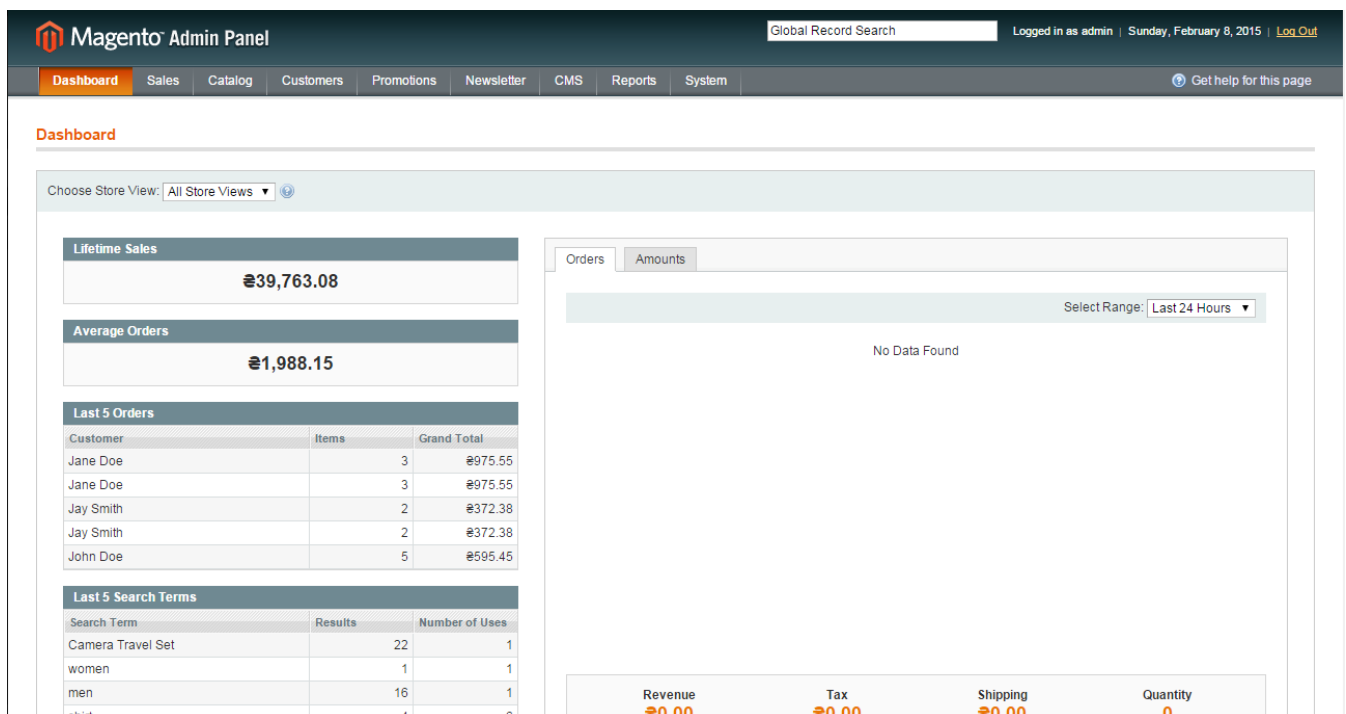


Рис.2.3. Головна сторінка адміністративної панелі

Головне меню містить інші пункти, які згруповані за призначенням.

Так в групі «Sales», тобто продажі, можна переглянути інформацію про замовлення, рахунки, доставку та оплату.

В групі каталог (Catalog) можна керувати товарами, категоріями товарів, атрибутами, пошуковими термами, переадресацією сторінок, оглядами та рейтингами.

В групі покупці (Customers) – можна керувати інформацією про покупців, групами покупців, здійснювати експорт та імпорт цих даних.

В групі акції (Promotions) – встановлювати різні акції та правила їх застосування.

В групі розсилок новин (Newsletter) – керувати розсилками новин та підписаними на ці розсилки користувачами.

В групі CMS – можна налаштовувати статичні сторінки, вміст статичних блоків, опитування та віджети.

В групі звіти (Reports) можна переглянути різні звіти та статистику діяльності інтернет-магазину.

В групі система (System) знаходяться різноманітні налаштування роботи системи.

2.5. Структура сайту магазину

В Magento веб-сайт (website) являє собою набір магазинів (stores), які в свою чергу є представленнями магазинів (store views). Ця ієрархія рівнів, можливо спочатку може збентежити, але саме ці рівні надають глибоку гнучкість налаштування онлайн-бізнесу в Magento.

Веб-сайт складається з одного чи більше магазинів, які розділяють між собою інформацію про клієнтів, інформацію про замовлення та торгову корзину.

Магазини є набором представлень та можуть бути налаштовані різними способами. Їх головна функція – забезпечити логічний контейнер для групування пов'язаних представлень магазину у веб-сайті.

Представлення є насправді екземплярами магазину в Magento. Більшість магазинів мають єдине представлення пов'язане з ними. Але магазин також може мати декілька представлень.

Наприклад, якщо вам потрібно відображати магазин на різних мовах, можна створити один магазин з кількома представленнями (рис. 2.4).

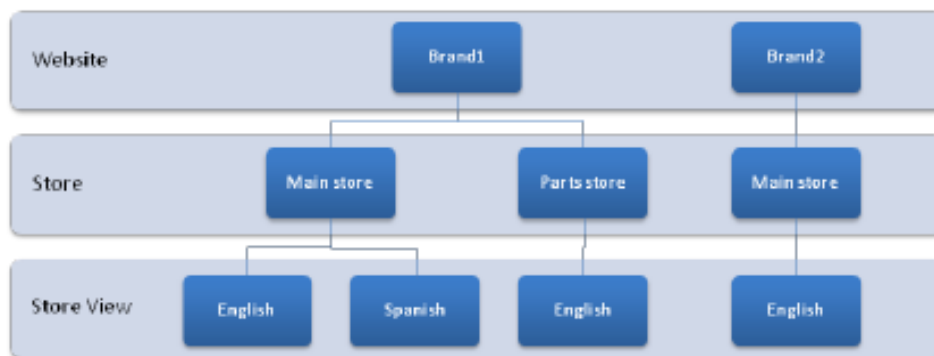


Figure 1. Hierarchy of websites, stores and store views in Magento.

Рис.2.4. Ієрархія магазину

Отже одна інсталяція Magento може використовуватись з кількома магазинами які можуть мати однаковий вигляд, але не обов'язково.

2.6. Теми дизайну

Для повного контролю над зовнішнім виглядом кожного магазину Magento дозволяє створювати теми оформлення (themes). Пов'язані теми групуються в пакети дизайну (design packages). Кожен магазин може використовувати власну тему, або вони можуть ділити одну, або створювати певну комбінацію.

Можна мати будь-яку кількість встановлених пакетів дизайну але не менше одного. При інсталяції Magento має спеціальний базовий пакет «base package» та інші пакети, в залежності від редакції. Тема може належати тільки одному пакету. Традиційно кожен пакет має містити «default theme» (тему за замовчуванням).

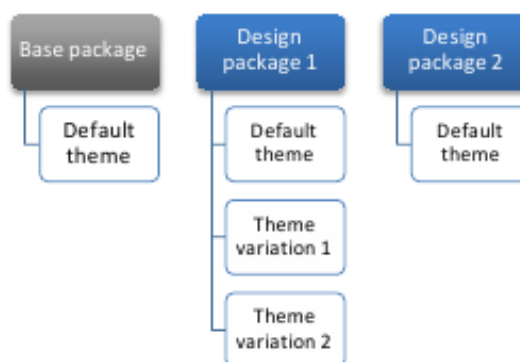


Рис.2.5. Ієрархія пакетів дизайну

Пакет дизайну та тема можуть бути закріплені як за веб-сайтом, так і за окремим магазином з адміністративної панелі Magento. Якщо призначити пакет на рівні веб-сайту, то всі магазини будуть наслідувати цей пакет. Це може бути перевизначено на рівні магазину призначенням іншого пакету.

Теми включають інформацію для шаблонування (розмітку, шаблони, переклад) та стильову інформацію (CSS файли, зображення, JavaScript сценарії). Ці файли знаходяться в двох головних директоріях:

- «App/design» – файли які визначають як шаблони сторінки виконуються.
- «Skin» – файли які контролюють візуальні аспекти теми – CSS, зображення, та ін.

Magento розподіляє файли таким чином для забезпечення більшого контролю над рівнем безпеки кожної директорії на сервері. Файли в «Skin» директорії мають бути доступними для веб-браузера. Файли в «App/design» директорії мають бути доступними тільки для додатку, і тому доступ до них ззовні може бути закритим. Всередині кожної з цих директорій файли розбиті по піддиректоріях в залежності від їхнього типу:

- «App/design»
 - Layout – Містить базові XML файли, які визначають блокову структуру різних сторінок та керують мета інформацією та кодуванням сторінки.
 - Template – Містить PHTML файли які включають XHTML розмітку та потрібний PHP для створення логіки для візуального представлення. Деякі шаблони є шаблонами сторінок, деякі – блоків.
 - Locale – Містить прості CSV текстові документи організовані в залежності від налаштованих мов та містять рядки з перекладом (пари ім'я–значення).
- «Skin»
 - CSS – Містить CSS файли, які використовуються для керування візуальним стилем веб-сайту.
 - Images – Містить всі зображення, які використовуються темою.

- JS – Містить специфічні для теми JavaScript-функції (більшість JavaScript бібліотек, які загальні для кількох тем, розміщуються в js/ в кореневій директорії).

Кожен пакет дизайну має містити тему з ім'ям «default», яка є головною темою для цього пакету. Коли пакет назначається на магазин, якщо не вказати тему, то буде автоматично обрана тема з ім'ям «default».

Архітектура тем в Magento була змінена між версіями Community Edition (CE) v1.3 і CE v1.4 та Enterprise Edition (EE) v1.7 і EE v1.8 для забезпечення простішої підтримки та «стійкості до оновлень» змінених тем. При цьому застосовується логіка відступу («fallback», рис. 2.6.) Наприклад, якщо в зміненій темі йде виклик файлу styles.css, але він відсутній, в цій зміненій темі буде здійснено пошук по ієрархії доки файл не буде знайдений. Це дозволяє в змінених темах зберігати тільки зміни відносно базової теми, а не копіювати усі файли.

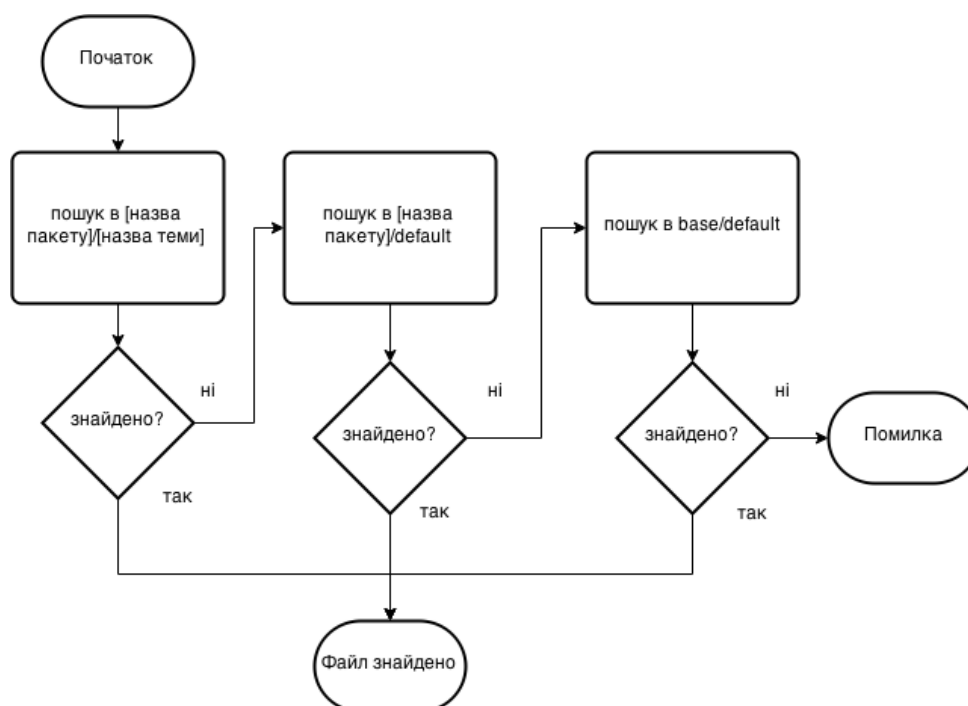


Рис. 2.6. Логіка відступу («fallback»)

2.7. Пакети та теми в структурі директорій Magento

Теми логічно згруповані в пакети дизайну, але файли тем розділені між двома директоріями. Назви директорій для пакетів дизайну та для тем мають бути однаковими в обох `app/design` та `skin` директоріях. Також зауважимо, що в кожній із директорій є необхідний набір директорій, в яких власне й зберігаються файли.

2.7.1. «Base» (базовий) пакет

Базовий пакет був доданий в CE v1.4 та EE v1.8. Призначенням базового пакету є забезпечення представлення для всього базового функціоналу. Таким чином користувацькі теми можуть включати в себе тільки зміни, що притаманні дизайну чи бізнесу, який вони мають підтримувати. Magento робить це використовуючи «fallback» модель, в якій пошук файлів, необхідних для відображення сторінки, проводиться спочатку в обраному користувацькому пакеті дизайну, а вже потім у базовому пакеті.

Базовий пакет є кінцевою точкою відступу для решти усіх пакетів і має бути незмінним при створенні чи редагуванні користувацьких тем. Це забезпечує чистіший код та кращий шлях оновлення для користувацьких тем тому що базовий пакет містить всі файли, які керують поведінкою Magento за замовчуванням, а користувацькі теми – тільки файли які були змінені. Базовий пакет містить тільки default тему і не варто додавати інші теми в базовий пакет.

В структурі директорій, базовий пакет має файли шаблонування та шкірки (skins), які знаходяться в `app/design` та `skin` директоріях відповідно. Базовий пакет містить тільки default тему, хоча це не повна тема, бо у ній не вистачає багатьох файлів шкірки.

2.7.2. Структура директорій базового пакету

app/design/frontend/base/default/	skin/frontend/base/default/
Містить усі файли розкладок та шаблонів необхідних для підтримки ядра функціоналу Magento.	Містить деякі CSS та JavaScript файли, які підтримують ядро функціоналу. Але не містить всі CSS файли та картинки необхідні для стилізації сайту, тому що вони відносяться до дизайну, а не до ядра.
<MAGENTO_BASE_DIR>	
+ app	
+ code	<MAGENTO_BASE_DIR>
+ design	+ skin
+ adminhtml	+ adminhtml
+ frontend	+ frontend
+ base	+ base
+ default	+ default
+ default	+ default
+ install	+ install
+ etc	
+ locale	
Mage.php	

Правила роботи з базовим пакетом:

- Не змінюйте файли в базовому пакеті.
- Не створюйте користувацькі теми в базовому пакеті.

2.7.3. Інші пакети за замовчуванням

В кожному дистрибутиві Magento міститься не тільки базовий пакет, а ще й принаймні один додатковий. В Community Edition цей пакет називається «default». В Professional Edition – «pro». В Enterprise Edition – «enterprise».

На відміну від базового пакету теми в цих демонстраційних пакетах є повними – з повним набором CSS файлів та картинок.

В Community Edition пакет default містить кілька тем:

app/design/frontend/default/

<MAGENTO_BASE_DIR>

+ app

+ design

+ frontend

+ base

+ default

+ **default**

+ blank

+ default

+ iphone

+ modern

skin/frontend/default/

<MAGENTO_BASE_DIR>

+ skin

+ frontend

+ base

+ default

+ **default**

+ blank

+ blue

+ default

+ iphone

+ modern

Professional Edition та Enterprise Edition містять тільки одну default тему в Pro та Enterprise пакетах відповідно.

2.7.4. Користувацькі теми оформлення

Для створення користувацької теми потрібно створити папки в користувацькому пакеті в app/design та skin директоріях.

Наприклад, якщо ви бажаєте створити пакет з назвою «Dubloo», то потрібно в кожній директорії фронтенду створити папку з відповідною назвою та папку «default» для теми в кожній з них.

app/design/frontend/dubloo/default/

<MAGENTO_BASE_DIR>

+ app

+ code

+ design

+ frontend

+ base

+ default

+ **dubloo**

+ default

skin/frontend/dubloo/default/

<MAGENTO_BASE_DIR>

+ skin

+ frontend

+ base

+ default

+ **dubloo**

+ default

Подальший вміст цих папок залежить від рівня змін відносно базового пакету.

Кожен пакет дизайну має містити хоча б «default»-тему та може мати будь-яку кількість варіацій.

app/design/frontend/dubloo/default/

<MAGENTO_BASE_DIR>

- + app
- + design
- + frontend
- + base
- + default
- + dubloo
- + default
- + layout
- + template

skin/frontend/dubloo/default/

<MAGENTO_BASE_DIR>

- + skin
- + frontend
- + base
- + default
- + dubloo
- + default
- + css
- + images
- + js

Є деяка відмінність при роботі з РЕ та ЕЕ від СЕ. Додаткова функціональність в РЕ та ЕЕ не є частиною базового пакету, а включена в pro та enterprise пакети відповідно. Для створення користувацького пакету для РЕ та ЕЕ, потрібно скопіювати default тему, з pro та enterprise пакетів відповідно, до default теми нового користувацького пакету. Тоді всі зміни робляться в варіації теми.

Альтернативним варіантом є створення користувацької теми в pro або enterprise пакетах, але рекомендовано створювати новий пакет для зменшення ризику зміни коду enterprise/default або pro/default.

app/design/frontend/dubloo/your_theme/

<MAGENTO_BASE_DIR>

- + app
- + design
- + frontend
- + base

skin/frontend/dubloo/your_theme/

<MAGENTO_BASE_DIR>

- + skin
- + frontend
- + base
- + enterprise

+ enterprise	+ dubloo
+ dubloo	+ default (copy from EE/def)
+ default (copy from EE/def)	+ your_theme
+ your_theme	+ css
+ layout	+ images
+ template	+ js

2.8. Застосування тем

Розглянемо як створити та застосувати нову тему дизайну та застосувати її до Magento магазину. Наприклад маємо новий пакет з деякими незначними змінами або завантажений з Magento Connect, або від іншого колеги.

Тепер для застосування теми перейдемо в адміністративну панель (наприклад, www.mydomain.com/admin), далі на вкладку налаштування дизайну (System -> Configuration -> Design tab). Якщо у вас більше одного веб-сайту, магазину чи представлення, то у верхньому лівому кутку лівої колонки буде блок з назвою «Current Configuration Scope» де можна обрати магазин до якого застосовуються зміни.

Крок 1.

З вкладки дизайну в поле «Current package name» введіть назву пакету, в якому знаходиться ваша нова тема. Якщо поле залишити пустим використовується «default» пакет.

Крок 2.

В default поле (в Themes панелі) введіть ім'я нової теми яку ви хочете застосувати до веб-сайту магазину чи представлення (рис. 2.7). Якщо залишити це поле пустим то буде застосовано «default»-тему з пакету, який задано раніше. Відповідно до схеми відступу в Magento, якщо задано ім'я теми, буде пошук файлів спочатку в заданій темі потім в «default» темі пакету, а в кінці у «default»-темі «base»-пакету. Можна задати, також, окремо тему для певних типів файлів таких як переклади, розкладки, шаблони та шкірки (skins).



Рис.2.7. Встановлення теми в адміністративній панелі

Крок 3.

Натисніть на «Save config» для збереження конфігурації та перезавантажте магазин.

Magento дозволяє гнучко використовувати кілька тем навіть в одній розкладці та шаблоні. Наприклад, нам потрібно мати однакову структуру шаблону та розкладки для всіх представлень, але різну графіку та кольорову схему. З Magento це вирішується просто:

Директорії розкладки та шаблону `app/design/frontend/[пакет]/[тема]` та власне стилів, які розташовані в `skin/frontend/[пакет]/[тема]`, не обов'язково мають мати однакову назву. Якщо поглянути на звичайну інсталяцію Magento, можна побачити default тему в `app/code/design/frontend/default/default`. Їй відповідають одразу декілька папок зі шкірками в `skin/frontend/default`: «blank», «blue», «default», «French» та «German». Всі п'ять є різними шкірками, які можуть використовуватись з default темою. Є також «blank», «iPhone» та «modern» теми. В директорії `skin` більше тем, ніж у директорії `app/design`, тому що «blue», «French» та «German» теми відрізняються від «default»-теми тільки CSS.

Припустимо, для нашого магазину як Default встановлено default тему. Таким чином, англійське представлення буде виглядати так (рис.2.8):



Рис.2.8. Фронтенд англійського представлення з темою default/default

Наприклад, потрібно щоб іспанські відвідувачі бачили різницю не тільки в мові, коли вони переключають представлення. Для досягнення цього можна призначити «blue» тему для іспанського представлення (рис.2.9 та рис.2.10).

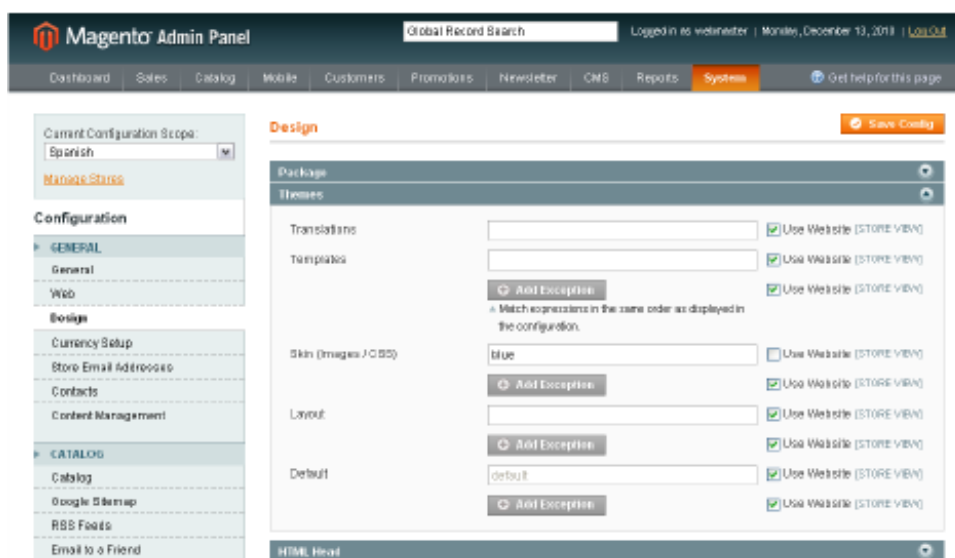


Рис.2.9. Встановлення default/blue теми для іспанського представлення



Рис.2.10. Фронтенд іспанського представлення з default/blue темою.

2.9. Виключення дизайну

Виключення дизайну дозволяють встановлювати альтернативні теми в залежності від user-agent. Замість того щоб створювати окремі представлення для різних пристроїв, наприклад смартфонів чи мобільних телефонів (рис.2.11).

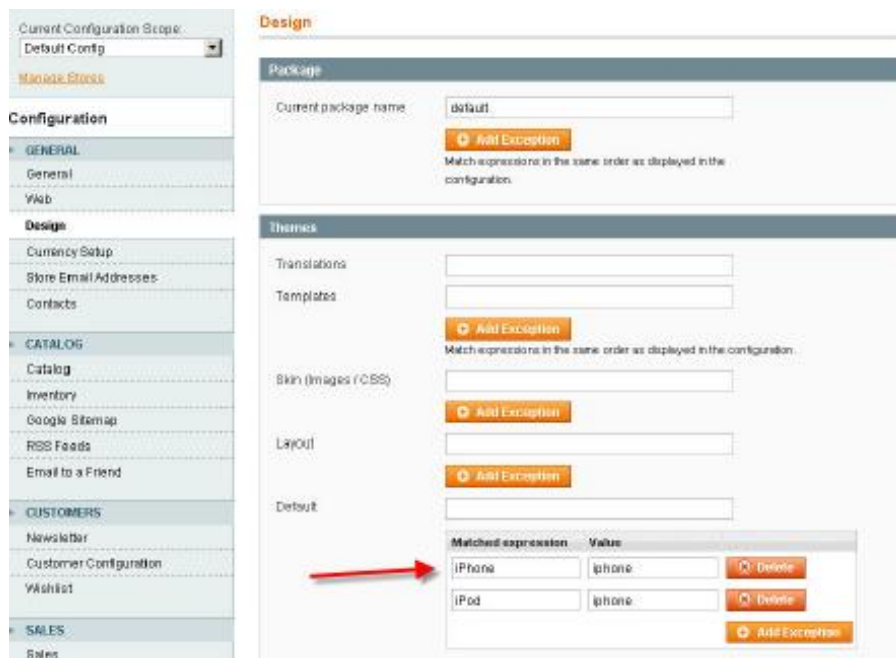


Рис.2.11. Використання виключень дизайну

2.10. Віджети

В Magento віджети – це фронтенд блоки з наперед визначеним набором конфігураційних опцій. Ці конфігураційні опції відображаються на спеціальній формі для редагування в бекенд панелі, коли віджет доданий до магазину. Здатність легко задавати конфігураційні опції віджета дає повний контроль над розміщенням віджету на сторінці для власника магазину. Важливим аспектом використання віджетів є те що при цьому знищуються деякі налаштування рівня теми, які зазвичай використовуються для налаштування магазину.

По суті, вони надають чудову можливість бізнес-користувачам без особливих технічних знань легко додавати динамічний контент (включаючи, наприклад, дані товарів), в місцях наперед передбачених дизайнером чи розробником. Це дозволяє краще та гнучкіше керувати створенням інформаційного та маркетингового контенту через інструменти адміністратора, дозволяючи інтуїтивно та ефективно керувати контентом таким як:

- Динамічні дані товарів маркетингової кампанії стартових сторінок.
- Динамічна інформація, така як нещодавно переглянуті товари в контентних сторінках.
- Рекламні зображення в різних блоках.
- Інтерактивні елементи та блоки (зовнішні системи рецензій, відео-чати, голосування та форми підписки).
- Альтернативні навігаційні елементи (хмара “тегів”, слайдер зображень каталогу тощо).
- Інтерактивні та динамічні Flash-елементи легко налаштовуються та вбудовуються в контентні сторінки для кращого сприйняття користувачами.

Термінологія віджетів

Фронтенд блок – це елемент який створює візуальний вихід або прив’язуванням візуальної структури, або продукуванням власне контенту.

Magento віджет – фронтенд блок який імплементує спеціальний інтерфейс віджета, що дозволяє мати різні конфігураційні опції для кожного екземпляру блоку та мати кілька незалежних екземплярів блоків на сторінках.

Екземпляр Magento-віджета – це блок на одній або кількох сторінках, який приймає свої конфігураційні опції, що задані власником магазину в бекенді. Один і той же віджет може бути доданий до фронтенду багато разів створюючи багато екземплярів одного віджета.

Один екземпляр Magento-віджета може бути також доданим на кілька сторінок (з однаковими конфігураційними опціями) і при цьому керуватись як одна сутність.

Приклади віджетів

Magento включає за замовчанням наступні віджети які можна налаштовувати. Нові віджети також можуть бути створені розробниками.

- CMS Page Link – відображає посилання на обрану сторінку системи керування контентом, та дозволяє задавати довільний текст та заголовок. Для цього віджету доступні два шаблони – строкове посилання та блок.
- CMS Static Block – відображає контент обраного статичного блоку.
- Catalog Category Link – відображає посилання на обрану категорію каталогу, та дозволяє задати довільний текст та заголовок. Доступні два шаблони: строковий та блочний.
- Catalog Product Link - відображає посилання на обраний товар каталогу, та дозволяє задати довільний текст та заголовок. Доступні два шаблони: строковий та блочний.
- Recently Compared Products – відображає блок, який містить недавно порівнювані товари. Цей віджет дозволяє задати кількість товарів які відображати та має два доступні шаблони (список та сітка).

2.11. Блоки

Блоки – це спосіб яким Magento відрізняє масив функціональності в системі та створює модульний спосіб управління як з візуального, так і з функціонального боку. Є два типи блоків, які працюють разом для створення візуального виходу.

Структурні блоки – це блоки, що створюються виключно для призначення візуальної структури для сторінок магазину, наприклад, шапка, низ, ліва колонка, головна колонка (рис.2.12).

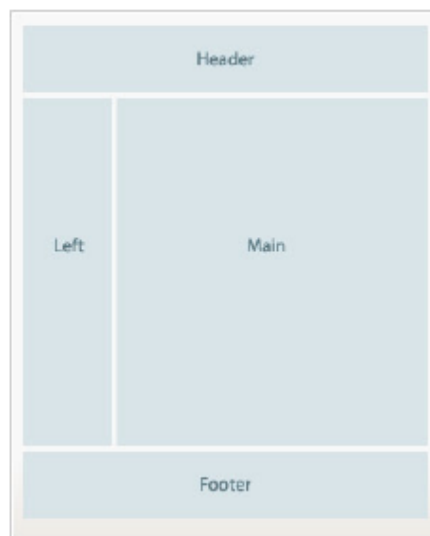


Рис. 2.12. Приклад структурних блоків.

Контентні блоки – це блоки для створення контенту в середині структурних блоків. Вони є представленням кожної функціональної можливості на сторінці та використовують шаблони для генерації xHTML для вставки в батьківський структурний блок. Наприклад, список категорій, міні-корзина, мітки товарів, списки товарів. Замість того, щоб підключати шаблон за шаблоном для того щоб зібрати цілісну сторінку на виході, як це роблять типові додатки електронної комерції, Magento збирає та розташовує вміст сторінки через блоки.

Діаграма класів блоків інтернет-магазину наведена на рис. 2.13.

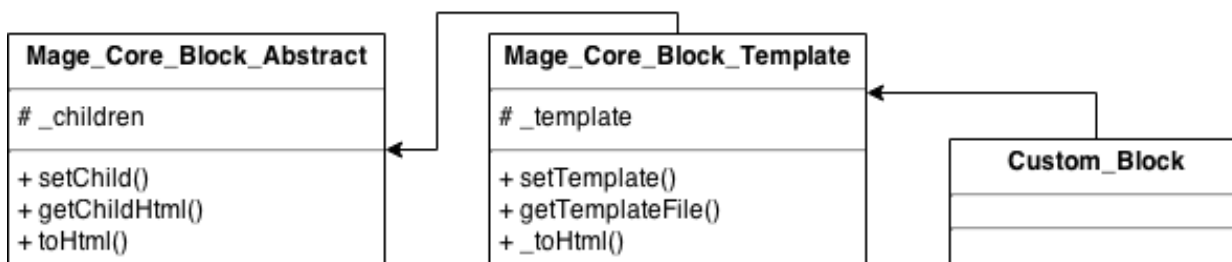


Рис. 2.13. Діаграма класів блоків

2.12. Розкладки

Розкладки (Layouts): ці файли по суті прив'язують контентні блоки до структурних блоків. Розкладки в Magento мають дві функції – вони визначають структурні та контентні блоки і повідомляють Magento: як та де їх з'єднувати (рис.2.14).



Рис.2.14. Приклад прив'язки блоків

Розкладка – це інструмент за допомогою якого можна призначати контентні блоки до структурних блоків. Розкладки в Magento – це XML файли. Змінюючи розкладку можна переміщати блоки по сторінці та призначати шаблони контентним блокам для створення розмітки структурних блоків.

Файли базової розкладки знаходяться в `app/design/frontend/base/default/layout`. Файли розкладки для користувацької теми мають бути в відповідній директорії: `app/design/frontend/<пакет>/<тема>/layout`.

Розкладка складається з default-розкладки та оновлень розкладки (layout updates). За допомогою цих оновлень можна легко змінювати зв'язки між контентними та структурними блоками і контролювати функції фронтенду, такі як завантаження чи вивантаження специфічного для блоку JavaScript на сторінці.

Кожен модуль Magento має свій власний файл розкладки, наприклад «`catalog.xml`» є файлом розкладки для модуля каталогу. Кожен файл розділений на ручки (handles). Доступні handles визначаються ядром Magento та активними доповненнями. Частіш за все, кожен handle відповідає типу сторінки в магазині, наприклад `<catalog_category_default>` та `<catalog_product_view>`, але є handles які застосовуються до всіх сторінок, такі як `<default>`, є й такі що відповідають певному статусу, наприклад `<customer_logged_in>` чи `<customer_logged_out>`.

Handle може зустрічатись в кількох файлах розкладок та блоки можуть бути призначені до цього handle чи переназначені кожним файлом розкладки. Наприклад, більшість файлів розкладки можуть містити `<default>` handle. Розбираючи файли розкладки, Magento спочатку бере оновлення розкладки призначені в `<default>` handle в кожному із файлів розкладки, читаючи їх в порядку, визначеному в `app/etc/modules/Mage_All.xml`. Тоді розбираються специфічні для сторінки layout handles, також переглядаючи всі файли розкладки для цього handle.

Система побудови сторінок в Magento зроблена, таким чином, для забезпечення вільного додавання та видалення модулів без впливу на інші модулі. Це означає, що можна додати, видалити чи перемістити більшість функціоналу простим додаванням, видаленням чи переміщенням блоку декларацій в файлі розкладки.

Розкладки містять малий набір XML-тегів які працюють в якості докладних команд для додатку, що кажуть як будувати сторінку, що має на ній бути та поведінку кожного будівельного блоку.

Handle – це ідентифікатор за яким додаток визначає що робити з оновленнями які в ньому знаходяться.

Якщо він називається <default>, тоді додаток знає що включені оновлення мають бути завантажені майже на кожній сторінці магазину перед завантаженням специфічних для сторінок розкладок (майже на кожній тому що деякі виняткові сторінки, такі як впливаюча картинка товару, не завантажують розкладку з <default> handle).

Magento визначає поведінку та візуальне представлення кожного будівельного блоку сторінки тег-`<block>`. Ми вже згадували про два типи блоків – структурні та контентні. Найкращий спосіб їх відрізнити це подивитись яка поведінка призначена цим блокам за допомогою атрибутів тегу. Структурні блоки зазвичай містять атрибут «as» через який додаток може спілкуватись з відповідною областю (використовуючи метод `getChildHtml`) в шаблоні. Default розкладка більшою мірою складається з структурних блоків таких як: «left», «right», «content» та «footer».

Для блоків доступні такі атрибути:

- `type` – це ідентифікатор класу модуля який визначає функціонал блоку.
- `name` – це назва блоку за якою інші блоки можуть посилатися на блок, в якому визначений цей атрибут.
- `before` та `after` – це два способи позиціонування контентних блоків в структурних блоках. `before="-"` та `after="-"` є командами які використовуються для позиціонування блоку відповідно з самого верху чи знизу структурного блоку.
- `template` – цей атрибут визначає шаблон, який буде представляти функціонал блоку в якому цей атрибут визначений. Наприклад, якщо в

цьому атрибуті зазначено 'catalog/category/view.phtml', додаток завантажить файл шаблону 'app/design/frontend/template/catalog/category/view.phtml'.

- action – використовується для керування функціоналом фронтенду, таким як завантаження та вивантаження JavaScript.
- as – ім'я за яким шаблон викликає блок, в якому цей атрибут визначений. Коли ви бачите PHP метод getChildHtml('block_name') який викликається з шаблону, він відноситься до блоку в якому атрибуту «as» присвоєно ім'я 'block_name'. Наприклад, метод `<?=$this->getChildHtml('header')?>` в шаблоні відноситься до `<block as="header">`.

<Reference> (посилання) використовується для посилання на інший блок. Роблячи посилання на інший блок оновлення всередині <reference> будуть застосовані до відповідного блоку.

Для посилання потрібно використовувати атрибут "name". Цей атрибут націлений на атрибут "name" тегу <block>. Таким чином посилання <reference name="right"> направлене на блок <block name="right"> (рис.2.15).

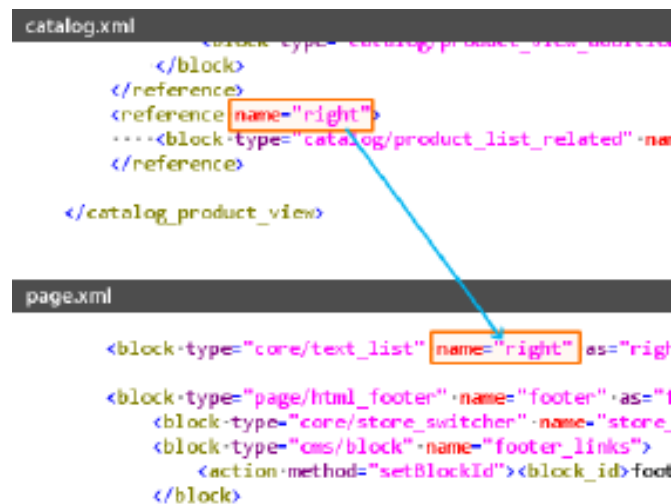


Рис.2.15. Приклад використання посилань.

Для швидшого знаходження потрібної розкладки можна ввімкнути підказки шаблонів в System -> Configuration -> Advanced -> Developer (рис.2.16).

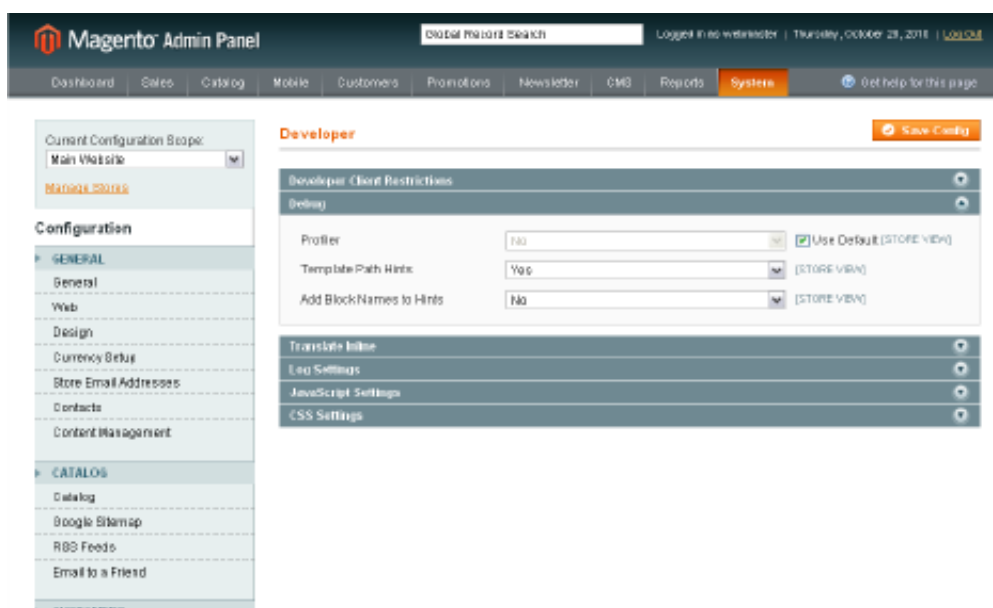


Рис.2.16. Підказки шаблонів в адміністративній панелі

Раніше ми розглядали як можна змінювати теми включаючи лише кілька змінених файлів у користувацьку тему. Це можливо завдяки механізму відступу. Цей метод зміни тем є абсолютно прийнятним. Але існує більш елегантний та легший для підтримки підхід ввімкнення та вимкнення блоків, який дозволяє зробити це, використовуючи лише один файл в директорії користувацької теми. Замість того щоб копіювати файли з базової теми та видаляти рядки коду, можна просто відмінити посилання на небажані блоки. Додатково, зберігання змін в одному файлі дозволяє простіше їх відслідковувати. За логікою Magento, після підключення та обробки усіх XML розкладок та їх оновлень, підключається локальний файл розкладки який називається «local.xml», він підключається останнім і за його допомогою можна перевизначити усі попередні оновлення розкладки. Файл «local.xml» має знаходитись в «app/design/frontend/<пакет>/<тема>/layout».

```
<?xml version="1.0" ?>
<layout>
    <default>
```

```

        <remove name="cart_sidebar" />
    </default>
</layout>

```

Зміст файлу буде залежати від базової теми, імена блоків для видалення можна подивитись у відповідному файлі розкладки. <Default> вказує на глобальну область, замість неї можна вказати конкретні структурні блоки такі як «left» чи «right».

«Local.xml» можна використовувати також для оновлення бажаних вузлів. Наприклад, для встановлення шаблону з 2-ма колонками зліва за замовчуванням, можна використати вузли, які керують відображенням сторінок каталогу, наприклад додавши такі рядки до файлу «local.xml»:

```

<catalog_category_default>
<reference name="root">
    <action method="setTemplate"><template>page/2columns-
left.phtml</template></action>
</reference>
</catalog_category_default>
<catalog_category_layered>
<reference name="root">
    <action method="setTemplate"><template>page/2columns-
left.phtml</template></action>
</reference>
</catalog_category_layered>

```

Використовуючи такий підхід можна також додавати блоки та змінювати їх положення на сторінці, а також їх поведінку, змінюючи лише один файл, та без потреби змінювати вихідні файли розкладки.

2.13. Шаблони

Тоді як файли розкладок керують присутністю чи відсутністю контентних блоків в темі, власне сам вміст цих блоків залежить від відповідних файлів шаблонів. Більшість шаблонів не містить логіки, яка визначає чи будуть вони відображені. Коли шаблон викликається, очікується що він буде розібраний та відображений.

Файли шаблонів в Magento є PHTML файлами які містять HTML та PHP, який буде розібраний та відображений відповідним браузером.

Взагалі рекомендується ті файли шаблонів, які мають бути змінені, копіювати в користувацьку тему з дещо зміненою назвою. Це зберігає оригінальний шаблон на випадок, якщо він використовується в декількох місцях.

```
<?xml version="1.0" ?>
<layout>
<catalog_category_default>
<reference name="root">
    <action method="setTemplate"><template>page/2columns-
custom.phtml</template></action>
</reference>
</catalog_category_default>
    <catalog_category_layered>
<reference name="root">
    <action method="setTemplate"><template>page/2columns-
custom.phtml</template></action>
</reference>
</catalog_category_layered>
</layout>
```

Розділ 3. Створення веб-сайту інтернет-магазину та власного модулю на базі CMS “Magento”

Для створення веб-сайту на базі Magento для початку необхідно отримати дистрибутив. Остання версія Magento Community Edition доступна для завантаження з офіційного сайту після безкоштовної реєстрації.

3.1. Встановлення системи на локальний сервер

Для розробки зазвичай використовується локальний сервер. Ми будемо використовувати WAMP, який містить в собі Apache, MySQL та PHP, заздалегідь налаштовані для комфортної розробки. Завантажимо з офіційного сайту останню версію Magento, в розпакованому вигляді. Також завантажимо, звідти ж, демонстраційні товари в вигляді SQL дамп-файлу та набору зображень. Помістимо розпакований дистрибутив в відповідну папку, розміщення якої залежить від налаштувань сервера. Створимо нову базу даних з назвою, наприклад, «magento». Тепер можна імпортувати демонстраційні товари в базу. Зображення потрібно помістити в папку «media». Налаштування бази можна провести за допомогою phpMyAdmin, що поставляється в комплекті з WAMP.

Після налаштування та запуску локального серверу можемо продовжити встановлення системи, відкривши задану адресу в браузері. На першій сторінці можна ознайомитись із ліцензією. Далі задаються налаштування локалізації, підключення до БД, базову адресу сайту, шлях до адміністративної панелі та ідентифікаційні дані адміністратора. На цьому початкові налаштування завершено та можна перейти до адміністративної панелі, або на головну сторінку сайту.

Кафедра КІТ (47)				НАУ 20 02 87 .000 ПЗ			
Виконала	Каменяр С.О.			СТВОРЕННЯ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ ТА ВЛАСНОГО МОДУЛЮ НА БАЗІ CMS “MAGENTO”	Літера	Аркуш	Аркушів
Керівник	Райчев І.Е.					52	25
Консульт					УС-201Мз 122 ⁵²		
Н.контр.	Райчев І.Е.						
.							

На даному етапі ми вже отримали фактично робочий магазин. Але для початку продажів потрібно наповнити його власними товарами, налаштувати способи оплати та доставки, а все це можна зробити з адміністративної панелі.

Тоді як адміністрування та наповнення контентом, завдяки зрозумілому інтерфейсу є не надто складними завданнями, зміна зовнішнього вигляду чи механізмів функціонування потребує більш глибокого розуміння системи.

3.2. Огляд реалізації схеми MVC

В типовому MVC (ModelViewController) фреймворку такому як Codeigniter, всі моделі знаходяться в одній директорії, всі контролери в іншій та всі представлення у своїй власній. Але в Magento це не так, вони згруповані в модулі, причому кожен модуль грає роль контейнеру для різного функціоналу. Модулі можуть складатися з моделі, контролера, блоку налаштувань (etc), допоміжних класів (Helper) та sql (опис таблиць бази даних та міграцій між версіями).

В Magento, щоб коректно використати модулі, їх спочатку необхідно підключити та налаштувати, виконавши відповідні описи в конфігураційних XML-файлах. На початку робочих дій читаються конфігураційні файли всіх активних модулів та об'єднуються в єдине дерево конфігурації. Далі йде розбір URL адреси, знаходиться відповідний модуль, інстанціюється відповідний контролер та викликається потрібний метод. Відповідність модулів певній адресі визначається шляхом звертання до конфігураційних файлів.

На відміну від інших MVC фреймворків, де контролер є відповідальним за отримання певних даних з моделі та передачі їх відповідному представленню (вигляду), в Magento контролери не обов'язково спілкуються з моделями, їх головною роллю є завантаження та побудова (render) розкладки без передачі даних. При цьому вигляди (view – інтерфейси, представлення) розділені на блоки, розкладки та шаблони.

Блоки є RHP-класами, які знаходяться в директорії модуля, шаблони є RHTML-файлами та знаходяться в директорії дизайну. Кожен блок прив'язаний до шаблону розкладкою.

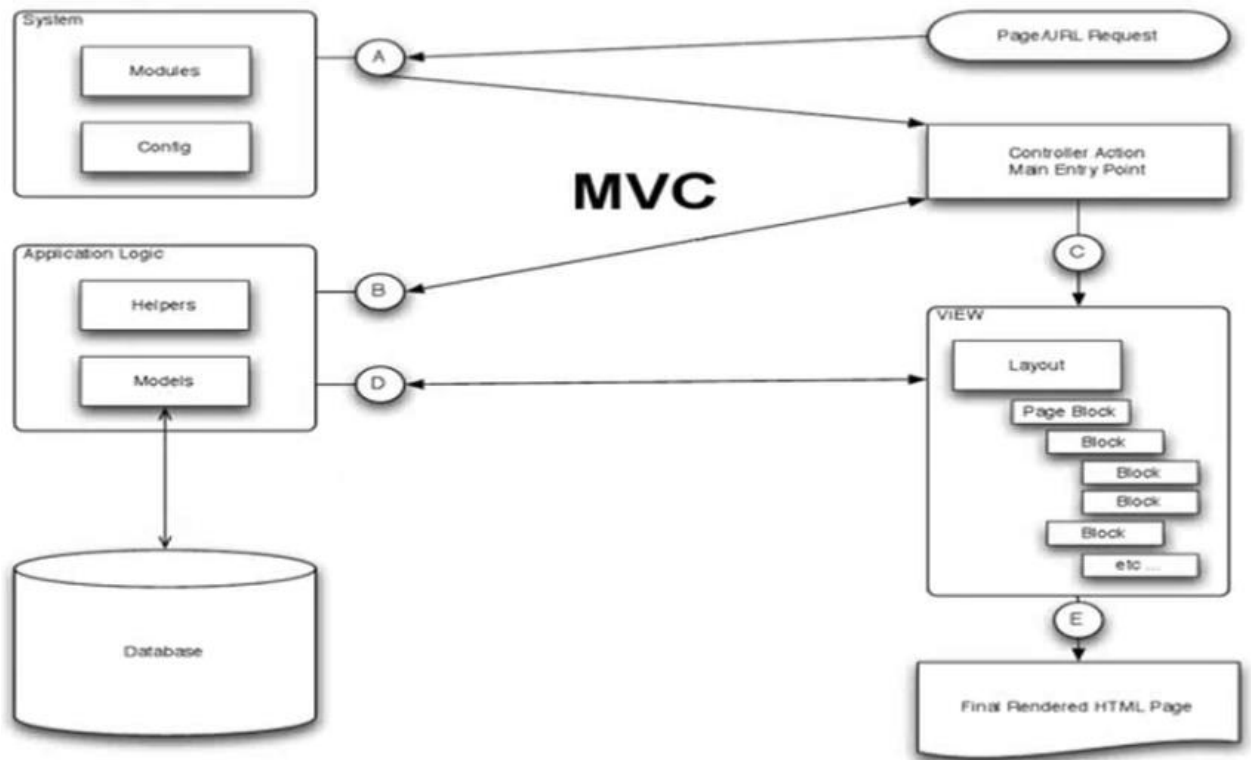


Рис. 3.1. Реалізація MVC в Magento

A. Розбір URL. Визначення контролера та дії.

С. Дія завантажує розкладку та запускає змалювання подання (представлення).

D. Шаблони через блоки отримують дані з моделей.

E. Блоки та дочірні блоки змальовуються в HTML сторінку.

3.3. Правила створення модулів

Структура розміщення модулів наступна: `/app/code/[область коду]/[namespace]/[назва модулю]`.

В «core» області коду знаходяться файли ядра системи, які не варто модифікувати, бо під час оновлення системи зміни можуть бути втрачені. Сторонні розробники мають зберігати свої файли, характерні для конкретної системи, в «local» області коду, а файли якими планується поділитись із спільнотою – в області «community». Namespace може бути довільним та часто відповідає назві компанії розробника.

Для прикладу створимо модуль. Розмістимо його в директорії `/app/code/local/Svetlana/Demo`. Для активації модуля потрібно створити конфігураційний XML файл в директорії `/etc/modules`. Назва конфігураційного файлу може бути довільною але традиційно він називається `[namespace]_[назва модулю].xml`, тому назвемо конфігураційний файл `Svetlana_Demo.xml`. Для активації модуля помістимо такий текст в щойно створений файл:

```
<config>
    <modules>
        <Svetlana_Demo>
            <active>true</active>
            <codePool>local</codePool>
        </Svetlana_Demo>
    </modules>
</config>
```

Після цього в адміністративній панелі на сторінці Configuration/Advanced з'явиться наш модуль.

Створимо файл `Product.php` в директорії `/app/code/local/Svetlana/Demo/Model`. В ньому буде розміщуватись клас відповідної моделі. Назва класу має бути `[namespace]_[назва модулю]_[тип`

класу]_[назва файлу], тобто `Svetlana_Demo_Model_Product`. Така назва необхідна для того щоб автозавантажувач зміг знайти потрібний клас.

Створимо новий метод, наприклад `sayHello`, який буде просто виводити деякий текст на друк. Для того щоб протестувати як працює наш модуль, створимо файл `test.php` з таким вмістом:

```
require_once 'app/Mage.php';
Mage::app();
$product = new Svetlana_Demo_Model_Product;
$product->sayHello();
```

Відкривши відповідну сторінку в браузері, переконаємось у тому, що дійсно викликався потрібний метод, так і відбулося завдяки автозавантажувачу який знаходиться в `/lib/Varien/Autoload.php`. Але в Magento для інстанціювання потрібних модулів краще використовувати іншу конструкцію.

3.4. Патерн «фабрика»

Абстрактна фабрика (Abstract Factory) — шаблон проектування, що забезпечує інкапсуляцію окремих фабрик у єдиній схемі, “приховуючи” їх деталізацію. Відноситься до класу породжуючих патернів проектування [12].

В типових випадках застосування, клієнтський код створює конкретну реалізацію абстрактної фабрики, а потім використовує загальний універсальний інтерфейс фабрики, для створення екземплярів об'єктів, які є частиною схеми.

Клієнтський код “не знає” (або не бере до уваги), які саме конкретно об'єкти він отримує від цих фабрик, оскільки він використовує універсальний інтерфейс для їх створення. Шаблон розмежовує деталі реалізації множини об'єктів від їх загального використання в коді, оскільки створення об'єкта здійснюється за допомогою методів, що забезпечуються інтерфейсом фабрики.

Слід використовувати шаблон *Абстрактна фабрика* коли:

- система не повинна залежати від того, як утворюються, компонуються та представляються об'єкти, що входять до неї;

- ті що входять до родини взаємозв'язані об'єкти повинні використовуватися разом і необхідно забезпечити виконання цього обмеження;
- система повинна конфігуруватися однією із родин об'єктів, що складають її;
- треба подати (представити) бібліотеку об'єктів, розкриваючи тільки їхні інтерфейси, але не реалізацію.

Діаграма класів абстрактної фабрики наведена на рис. 3.2.

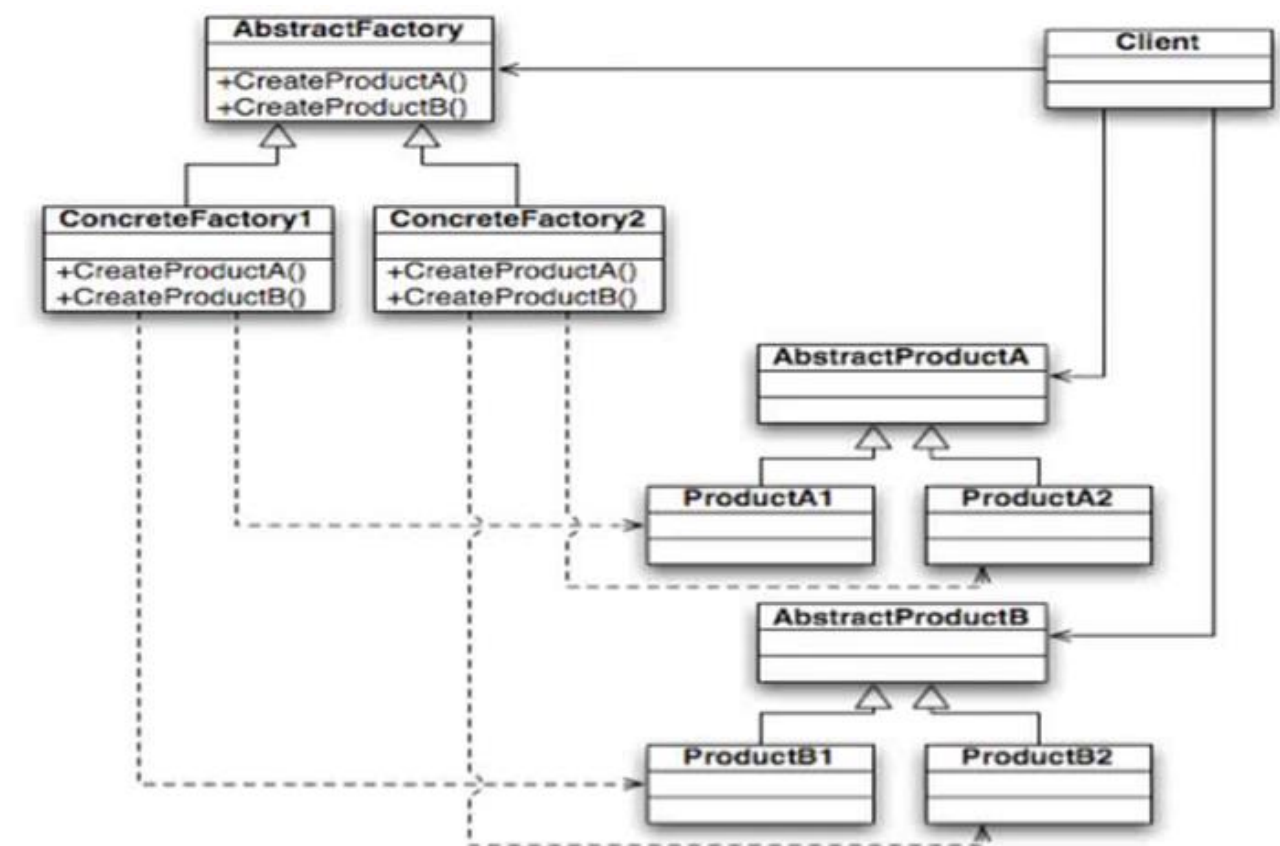


Рис. 3.2. Абстрактна фабрика

- **AbstractFactory** — абстрактна фабрика:
 - оголошує інтерфейс для операцій, що створюють абстрактні об'єкти-продукти;
- **ConcreteFactory** — конкретна фабрика:
 - реалізує операції, що створюють конкретні об'єкти-продукти;

- **AbstractProduct** — абстрактний продукт:
 - оголошує інтерфейс для типу об'єкта-продукту;
- **ConcreteProduct** — конкретний продукт:
 - визначає об'єкт-продукт, що створюється відповідною *конкретною фабрикою*;
 - реалізує інтерфейс *AbstractProduct*;
- **Client** — клієнт:
 - користується виключно інтерфейсами, котрі оголошенні у класах *AbstractFactory* та *AbstractProduct*.

Зазвичай під час виконання створюється (інстанціюється) єдиний екземпляр класу *ConcreteFactory*.

Ця конкретна фабрика створює об'єкти продукти, що мають досить визначену реалізацію. Для створення інших видів об'єктів клієнт повинен користуватися іншою конкретною фабрикою.

AbstractFactory передоручає створення об'єктів продуктів своєму підкласу *ConcreteFactory*.

Патерн «Фабрика» дозволяє гнучко змінювати функціонал за замовчуванням за допомогою конфігураційних файлів. Наприклад, замість `$customer=new Mage_Customer_Model_Session` використовується така конструкція `$customer = Mage::getModel("customer/session")`.

При цьому система знаходить потрібний файл завдяки конфігураційному дереву в якому записана назва директорії з класами моделі. Таким чином, змінивши конфігураційне дерево, можна змусити систему підключати користувацькі класи без необхідності вносити зміни в ядро системи.

Для того щоб `Mage::getModel()` зміг підключати нашу модель створимо файл `/app/code/local/Svetlana/Demo/etc/config.xml` з таким вмістом:

```
<config>
    <global>
        <models>
            <demo>
                <class>Svetlana_Demo_Model</class>
            </demo>
        </models>
    </global>
</config>
```

```
        </models>
    </global>
</config>
```

Тепер виклик `Mage::getModel('demo/product')` еквівалентний `new Svetlana_Demo_Model_Product`.

Таким же чином у конфігураційних файлах описуються блоки та хелпери (helpers) в відповідних вузлах конфігураційного дерева (blocks та helpers).

Створимо хелпер `Svetlana_Demo_Helper_Customer`, який буде знаходитись в `/app/code/local/Svetlana/Demo/Helper/Customer.php`. Часто користувацькі хелпери наслідуються від `Mage_Core_Helper_Abstract`.

На відміну від моделей та блоків хелпери викликаються не через `getModel` чи `getBlock`, а через `Mage::helper`, наприклад для нашого хелпера виклик буде `Mage::helper('demo/customer')`.

Під час виклику хелперів є ще одна особливість: якщо не вказувати назву файлу, наприклад `Mage::helper('demo')`, то буде викликаний клас `Svetlana_Demo_Helper_Data`.

Наприклад, змінимо метод `getChildren()` в `Mage_Catalog_Model_Category`, цей метод повертає прямих нащадків категорії як рядок назв, розділених комами, а нам, наприклад, потрібно отримати масив.

Не варто змінювати файл ядра, тому що зміни можуть бути втрачені при оновленні системи. Виходимо з того, що Magento підключає файли у такому порядку: `local`, `community`, `core`.

Тому можна скопіювати `/app/code/core/Mage/Catalog/Model/Category.php` у `/app/code/local/Mage/Catalog/Model/Category.php` і вносити зміни вже там. Але при оновленні системи можуть бути додані нові функції та виправлені помилки в існуючих.

Є інший метод зміни функціоналу ядра, який дозволяє отримати ці оновлення. Для цього потрібно створити власний клас моделі, наприклад, `Svetlana_Catalog_Model_Category`, який буде наслідувати клас ядра.

Для того щоб система підключала наш клас замість класу ядра потрібно у файлі `etc/config.xml`, який знаходиться в нашому модулі прописати:

```
<config>
  <global>
    <models>
      <catalog>
        <rewrite>
          <category>Svetlana_Catalog_Model_Category</category>
        </rewrite>
      </catalog>
    </models>
  </global>
</config>
```

Таким же чином можна модифікувати також блоки і хелпери.

3.5. Події

Інший важливий інструмент який дозволяє змінювати поведінку системи за замовчуванням – події (events). На події посилаються в багатьох місцях системи.

Подійно-орієнтоване програмування (англ. event-driven programming; ПОП) – це парадигма програмування, в якій виконання програми визначається подіями: діями користувача (клавіатура, миша), повідомленнями інших програм і потоків, подіями операційної системи (наприклад, надходженням мережевого пакета).

ПОП можна також визначити як спосіб побудови комп'ютерної програми, при якому в коді (як правило, в головній функції програми) явним чином виділяється головний цикл програми, тіло якого складається з двох частин: отримання повідомлення про подію і процес обробки події.

Як правило, в реальних завданнях виявляється неприпустимим тривале виконання обробника події, оскільки при цьому програма не може реагувати на інші події. У зв'язку з цим при написанні подійно-орієнтованих програм часто застосовують автоматне програмування.

Подійно-орієнтоване програмування, як правило, застосовується в трьох випадках:

- під час побудови користувацьких інтерфейсів (у тому числі графічних);
- під час створення серверних застосунків (додатків) у разі, якщо з тих чи інших причин небажано породження обслуговуючих процесів;
- під час програмування ігор, в яких здійснюється управління значною кількістю об'єктів.

Подієво-орієнтоване програмування застосовується в серверних програмах для вирішення проблеми масштабування на 10.000 одночасних з'єднань і більше.

У серверах, побудованих за моделлю «один потік на з'єднання», проблеми з масштабованістю виникають з наступних причин:

- занадто великі накладні витрати на структури даних операційної системи, необхідні для опису однієї задачі (сегмент стану завдання, стек і т.д.);
- занадто великі накладні витрати на перемикання контекстів.

Філософською передумовою для відмови від потокової моделі серверів може служити вислів Алана Кокса: «Комп'ютер – це скінченний автомат. Потокове програмування потрібно тим, хто не вміє програмувати скінченні автомати» [13].

Серверний застосунок при подійно-орієнтованому програмуванні реалізується на основі системного виклику, який отримує повідомлення події одночасно від багатьох дескрипторів (мультиплексування). При обробці подій використовуються виключно неблокуючі операції введення-виведення, щоб ні один дескриптор не перешкоджав обробці подій від інших дескрипторів.

Краще використовувати події, ніж наслідувати класи, якщо це дає можливість досягти потрібного результату, тому що при цьому зберігається підтримка при оновленні.

Для реєстрації спостерігача (observer), який буде слухати появу події, потрібно внести запис у конфігураційне дерево. У файлі конфігурації модуля створюється вузол, наприклад для події «customer_login»:

```
<config>
    <global>
        <events>
            <customer_login>
                <observers>
                    <demo>
                        <type>model</type>
                        <class>demo/observer</class>
                        <method>logCustomer</method>
                    </demo>
                </observers>
            </customer_login>
        </events>
    </global>
</config>
```

Тип може бути model або singleton.

Синглтон в Magento дещо відрізняється від інших. При виклику методу `getSingleton()` повертається один і той же об'єкт, але разом з тим можна отримати інші об'єкти шляхом виклику методу `getModel()`.

3.6. Модель бази даних

Існує різниця між Magento та іншими CMS в тому, як зберігаються дані. Дизайн бази даних досить унікальний. БД містить близько 300 таблиць.

Загалом моделі використовуються для зв'язку з базою даних, представляючи дані в об'єктно-орієнтованому виді та виконуючи бізнес логіку.

Для більшої абстракції Magento використовує тришаровий підхід до моделей, тобто маємо: моделі сутностей, моделі ресурсів та колекції.

Модель сутності виконує певну бізнес логіку, але також надає доступ до CRUD (create, read, update, delete) операцій, що насправді досягається через модель ресурсів. Таким чином моделі сутностей не містять речей пов'язаних із базою даних, таких як з'єднання чи посилення запитів до БД, замість цього кожна модель використовує відповідну модель ресурсів, яка спілкується з БД через адаптер та виконує всі CRUD завдання [14].

Розділяючи логічну модель (сутність) та код який спілкується з БД (модель ресурсів та адаптери), можливо писати нові адаптери для різних БД при незмінних моделях, а це і є найбільшою перевагою такого підходу.



Рис.3.3. Тришарова модель

Існують різні типи моделей.

В простих моделях кожен запис в БД є окремою сутністю. Наприклад, в таблицях конфігурації чи URL Rewrite, де кожний рядок представляє повну інформацію про сутність. Такий підхід часто використовується в інших системах.

Є й більш складні моделі такі як сторінки CMS, або ще складніші, такі як котирування продажів та замовлення. В них для отримання повного запису сутності потрібно переглянути кілька таблиць.

Також існує EAV (entity-attribute-value, сутність-атрибут-значення) моделі. Діаграма класів наведена в Додатку А. Такі моделі використовуються коли таблиця представлення є розрідженою матрицею. Розріджена матриця — це матриця, більша частина елементів якої є нулі. Прикладами таких моделей є товари, категорії та покупки. Вони найбільш гнучкі, але з деякими накладними витратами. Товари та категорії зберігаються майже в 50 таблицях. Тому EAV є одною з найбільш складних частин системи. Але не обов'язково знати подробиці,

завдяки Magento ORM (object-relational mapping, об'єктно-реляційне відображення), можна працювати з БД використовуючи об'єкти.



Рис.3.4. Типи моделей

EAV модель.

- Entity – сутність яка моделюється, наприклад товар чи категорія.
- Attribute – атрибут, властивість сутності.
- Value – значення атрибуту.

Розглянемо звичайну таблицю користувачів [14] (Таблиця. 2.1).

Таблиця. 2.1. Звичайна таблиця користувачів

customer_id	username	Password	first_name	last_name
1	milanche	121hjfd14588...	Milan	Stojanov
2	t_anja	5484dsd8464...	Tanja	Simic
3	vstojlkovic	Sde454adf45...	Vladimir	Stojlkovic

Припустимо нам потрібно мати велику кількість колонок для користувачів але не всі з них будуть заповненими для кожного з них. В результаті отримаємо таблицю з великою кількістю пустих значень. EAV вирішує цю проблему. Наведену вище таблицю можна замінити EAV з трьома таблицями:

Таблиці. 2.2. – 2.4. EAV таблиці користувачів

customer_id	username	password
1	milanche	121hjfd14588...
2	t_anja	5484dsd8464...
3	vstoiljkovic	Sde454adf45...

entity_id	attribute_id	value
1	1	Milan
1	2	Stojanov
2	1	Tanja
2	2	Simic
3	1	Vladimir
3	2	Stoiljkovic

attribute_id	Name	display	type
1	first_name	First	varchar
2	last_name	Last	varchar

entity_id	attribute_id	value
1	1	Milan
1	2	Stojanov
2	1	Tanja
2	2	Simic
3	1	Vladimir
3	2	Stoiljkovic

В першій таблиці маємо базові дані, які є обов'язковими. В таблиці атрибутів зберігається свого роду метаінформація про сутності. В ній зберігається атрибути які кожна сутність може мати. При цьому додавання нового атрибуту зводиться до простого додавання нового рядка до таблиці атрибутів. В таблиці значень зберігаються значення певних атрибутів для певних сутностей. Для кожного типу значень існує своя окрема таблиця.

В деякий модулях можна знайти директорію Mysql4, вона знаходиться там для зворотної сумісності, тому що підтримка кількох БД була додана лише в версії 1.6. До цього в моделях ресурсів був присутній специфічний для MySQL код. Тепер в них знаходиться лише код який не залежить від БД, а всі специфічні запити формуються в адаптерах.

Моделі ресурсів налаштовуються в тому ж вузлі конфігураційного дерева, що й відповідна модель ресурсів, поруч з її назвою класу, вузол з назвою іншого вузла в якому і вказаний клас моделі ресурсів.

```

<cms>
  <class>Mage_Cms_Model</class>
  <resourceModel>cms_resource</resourceModel>
</cms>
<cms_resource>

```

```

<class>Mage_Cms_Model_Resource</class>
<deprecatedNode>cms_mysql4</deprecatedNode>
<entities>
    <page>
        <table>cms_page</table>
    </page>
    <page_store>
        <table>cms_page_store</table>
    </page_store>
    <block>
        <table>cms_block</table>
    </block>
    <block_store>
        <table>cms_block_store</table>
    </block_store>

```

В той час як `Mage::getModel('cms/page')` відноситься до `Mage_Cms_Model_Page`, `Mage::getResourceModel('cms/page')` відноситься до `Mage_Cms_Model_Resource_Page`.

Але при цьому для зв'язування моделі з її ресурсною моделлю потрібно викликати метод `_init('cms/page')`, який зазвичай викликається в конструкторі об'єкту та визначений в `Mage_Core_Model_Abstract`, від якого більшість класів моделей наслідуються. Метод з такою ж назвою є в ресурсній моделі, під час його запуску читається конфігураційне дерево, в якому прописані назви таблиць в БД.

Завдяки Magento ORM отримувати дані з БД можна в зручний спосіб, наприклад отримаємо деякі дані товару:

```

$product = Mage::getModel('catalog/product')->load(42);
$metaTitle = $product->getData('meta_title');
$product->setData('meta_title', 'updated');
$product->save();

```

Для отримання даних одразу про кілька товарів використовуються колекції. Колекції не потрібно описувати в конфігурації, але потрібно створити файл `Resource/[назва ресурсу]/Collection.php`, який містить відповідно модель колекції, наприклад для відповідного товару `/app/code/core/Mage/Catalog/Model/Resource/Product/Collection.php`.

Для отримання назв товарів, наприклад:

```

$products = Mage::getModel('catalog/product')->getCollection()
    ->addAttributeToSelect(array('name', 'price'))
    ->addFieldToFilter('price', array('lt'=>50));
echo count($products) . '<br>';
foreach($products as $product){

```

```

echo $product->getData('name');
echo '<br>';
}

```

Діаграма класів колекції наведена на рисунку 3.5 [15-18].

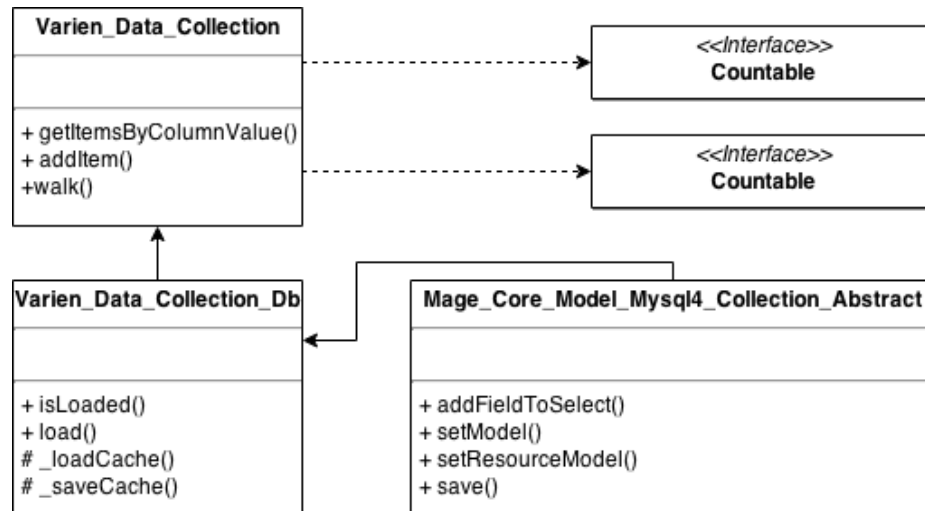


Рис. 3.5. Діаграма класів для колекцій

Слід зазначити, що без виклику над об'єктом колекції методу «`addAttributeToSelect`» повертається тільки вибірка по базовій таблиці EAV. Для збільшення швидкості рекомендується включати до вибірки лише потрібні атрибути.

3.7. Адресація та контролери

Звичайно всі запити перенаправляються на `index.php` який знаходиться в кореневій директорії інсталяції Magento.

Діаграми послідовності [15-18], які відображають процес обробки запиту браузера, наведені на рис.3.6 та рис.3.7.

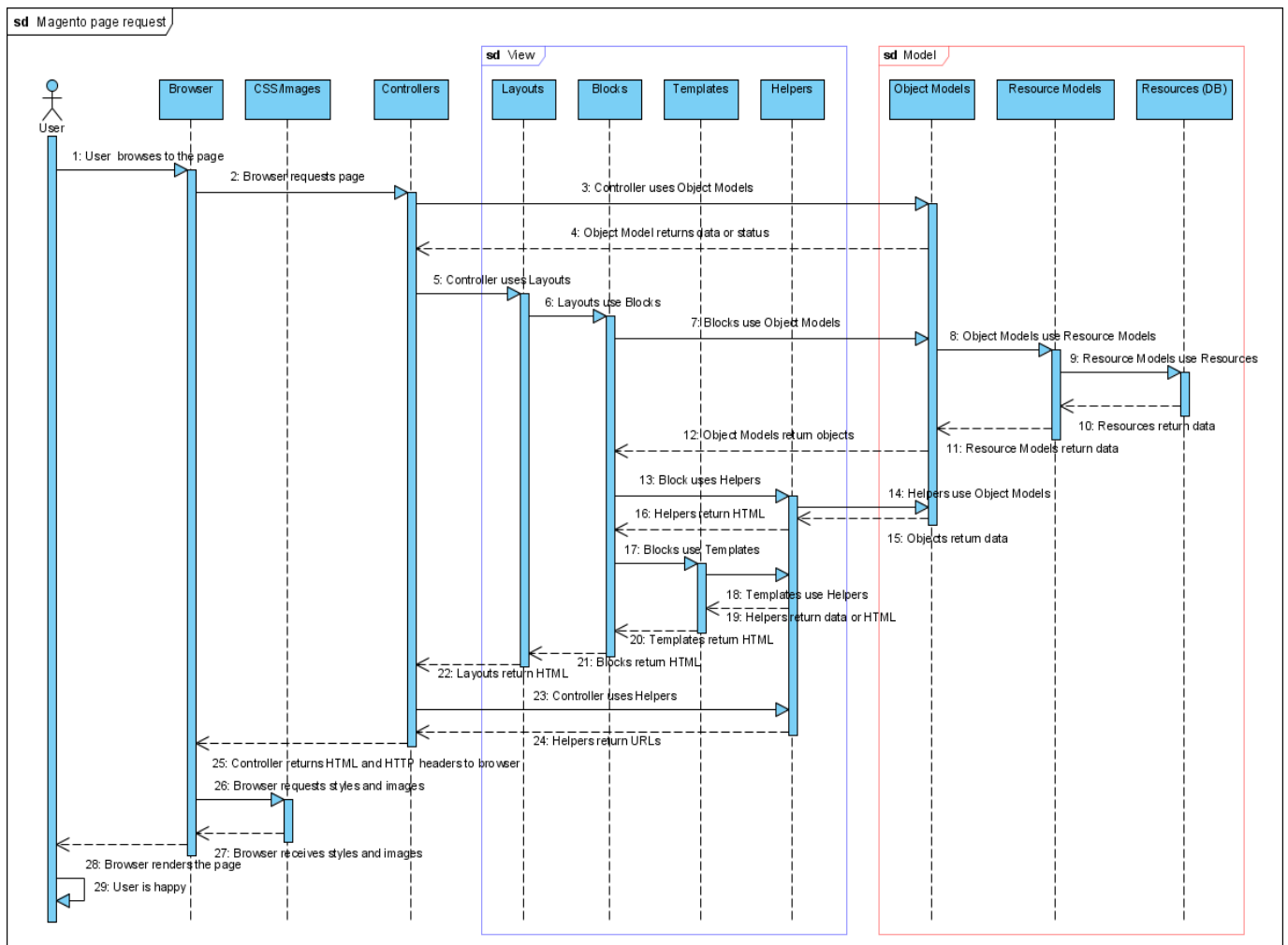


Рис. 3.6. Діаграма послідовності обробки запиту браузера

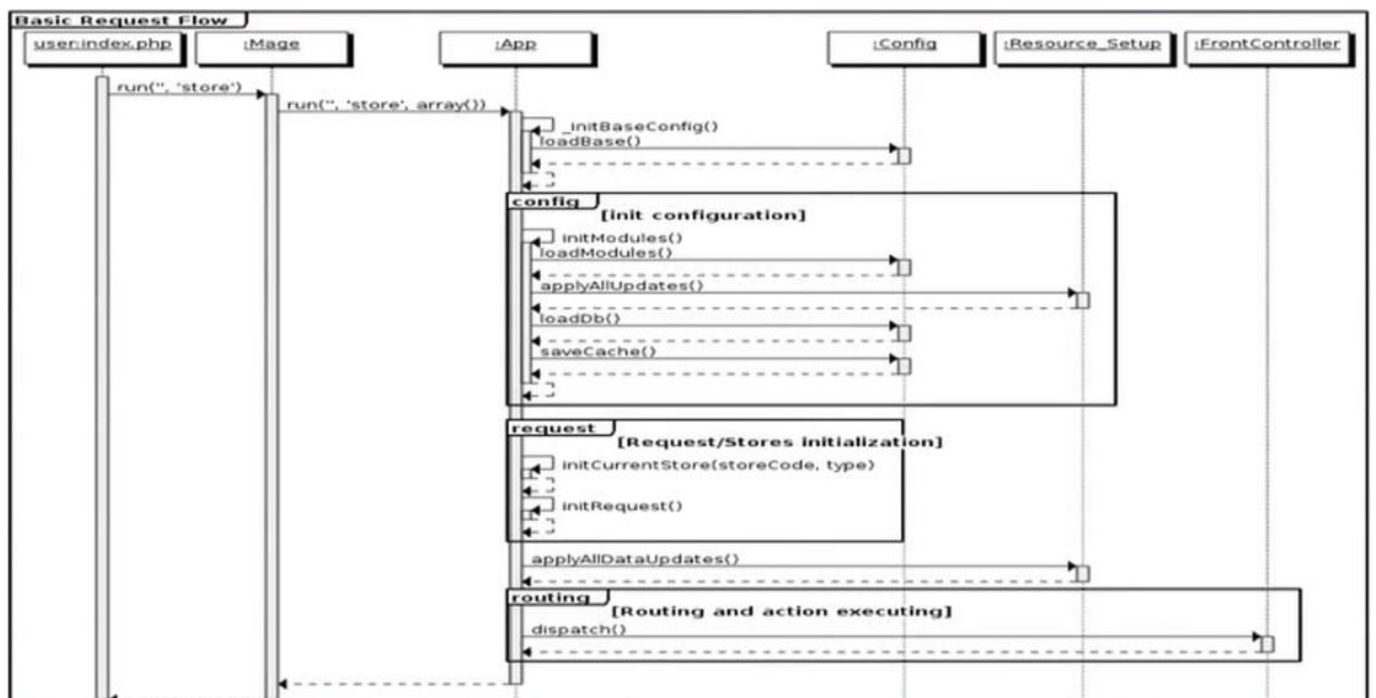


Рис. 3.7. Діаграма послідовності обробки запиту браузера

Якщо не задана конкретна адреса, а іншими словами відбувся запит домашньої сторінки сайту, запит оброблює контролер, який визначений в налаштуваннях. Ці налаштування можна змінити в адміністративній панелі на сторінці System/Configuration/Web на вкладці Default Pages.

Для виклику певного контролера використовуючи стандартний маршрутизатор, потрібно здійснити запит за адресою [адреса сайту]/index.php/[frontName]/[назва контролеру]/[назва дії]. Якщо назва контролеру чи назва дії не вказані, замість них підставляється index. Кожному модулю можна задати frontName в конфігураційному вузлі <outers> <[унікальна назва]> <args> <frontName>, наприклад:

```
<outers>
  <Svetlana_demo>
    <use>standard</use>
    <args>
      <module>Svetlana_Demo</module>
      <frontName>demo</frontName>
```

За замовчуванням контролером домашньої сторінки є «cms».

Крім стандартного маршрутизатора можуть бути задані й інші. Так можна мати зручну адресу, наприклад, для сторінки товару – [адреса магазину]/[назва товару].

Контролер може отримати передані із запитом дані за допомогою методу getRequest().

Найчастіше все що відбувається в контролері – це виклик методів loadLayout() та renderLayout() для завантаження та побудови розкладки.

В залежності від умов, таких як адреса сторінки, назва представлення магазину, тип користувача та інших, під час завантаження розкладки генеруються якорі розкладки (layout handles), які потім використовуються для вибору вузлів розкладки, що потрібно побудувати та відобразити.

Під час побудови розкладки (renderLayout) підключаються необхідні блоки та шаблони, що необхідні для відображення.

Розкладки складаються в основному з блоків. Кожен блок в розкладці має два обов'язкових атрибуту тип (type) та ім'я (name). Тип вказує на назву класу блоку, ім'я використовується для подальшого посилання на блок. Існує спеціальний блок з ім'ям «root» тобто кореневий блок, він зазвичай має атрибут «output» зі значенням «toHtml», який вказує системі, що блок має бути виведений браузеру.

Шаблон можна підключити до блоку в самому блоці, або використовуючи спеціальний атрибут «template», при цьому вказуючи його значенням шлях до шаблону відносно директорії теми.

Кожен шаблон представляє собою HTML з деякими вкрапленнями викликів PHP методів, таких як «`$this->getChildHtml('head')`». «`$this`» вказує на поточний блок, «`head`» відповідає значенню атрибута «`as`» одного з дочірніх блоків в розкладці. Таким чином блоки викликають інші блоки, поки сторінка не буде побудована.

Є й інші блоки які не використовують шаблонів – текстові списки (text_list). У цих блоків автоматично будуються всі дочірні блоки. Вони є свого роду контейнерами для інших блоків. До таких блоків відносяться «left», «right» та «content». Автоматична побудова дочірніх блоків дає змогу посилатися на ці блоки із своїх файлів оновлення розкладки, причому без необхідності модифікувати файл шаблону батьківського блоку. Таким чином, включення чи виключення модуля з розкладкою, підключеною таким способом, не вплине на працездатність сайту. Іншим дочірнім елементом блоку може бути «action» (дія) з атрибутом «method» (метод), який використовується для виклику методів блоку.

3.8. Створення модулю

Для того щоб показати як все працює разом побудуємо модуль, який буде відображати всі товари що конфігуруються. Наш модуль матиме назву «Configurable».

Активуємо модуль помістивши, файл `Svetlana_Configurable.xml` в `/app/etc/modules`.

```
<config>
    <modules>
        <Svetlana_Configurable>
            <active>true</active>
            <codePool>local</codePool>
        </Svetlana_Configurable>
    </modules>
</config>
```

У локальному файлі конфігурації `/app/code/local/Svetlana/Configurable/etc/config.xml` встановимо:

- «frontName» значення «configurable».
- Директорії класів:
 - Для блоків – «Svetlana_Configurable_Block»
 - Для хелперів – «Svetlana_Configurable_Helper»
- Файл оновлення розкладки – `Svetlana_configurable.xml`

```
<config>
    <global>
        <blocks>
            <configurable>
                <class>Svetlana_Configurable_Block</class>
            </configurable>
        </blocks>
        <helpers>
            <configurable>
                <class>Svetlana_Configurable_Helper</class>
            </configurable>
        </helpers>
    </global>
    <frontend>
        <routes>
            <Svetlana_configurable>
                <use>standard</use>
                <args>
                    <module>Svetlana_Configurable</module>
                    <frontName>configurable</frontName>
                </args>
            </Svetlana_configurable>
        </routes>
        <layout>
            <updates>
                <configurable>
                    <file>Svetlana_configurable.xml</file>
                </configurable>
            </updates>
        </layout>
    </frontend>
</config>
```

Створимо клас «Svetlana_Configurable_Block_Configurable» з методом «getConfigurableProducts», який буде повертати колекцію товарів із ідентифікатором типу, рівним «configurable».

```
<?php
class Svetlana_Configurable_Block_Configurable extends Mage_Core_Block_Template {

    public function getConfigurableProducts(){
        $products = Mage::getModel('catalog/product')
            ->getCollection()
            ->addAttributeToSelect(array('name', 'price', 'url_key'))
            ->addAttributeToFilter('type_id', array('eq' =>
'configurable'));
        return $products;
    }
}
```

Створимо індексний контролер з індексною дією, яка буде просто завантажувати та будувати розкладку.

```
<?php
class Svetlana_Configurable_IndexController extends
Mage_Core_Controller_Front_Action{
    public function indexAction(){
        $this->loadLayout();
        $this->renderLayout();
    }
}
```

Створимо хелпер «Svetlana_Configurable_Helper_Data» з методом «beautifyPrice» для форматування виводу цін.

```
<?php
class Svetlana_Configurable_Helper_Data
    extends Mage_Core_Helper_Abstract {

    public function beautifyPrice($price){
        if(!isset($price)){
            return 'unknown price';
        }
        return number_format($price, 2, '.', ',');
    }
}
```

Створимо варіацію теми та ввімкнемо її в адміністративній панелі.

В директорії щойно створеної теми:

- Створимо файл розкладки «layout/Svetlana_configurable.xml», в якому для якоря «Svetlana_configurable_index_index», що відповідає індексній сторінці нашого контролера, додамо дочірній блок до блоку «content», призначаючи йому шаблон «Svetlana/configurable.phtml».

```
<layout>
    <Svetlana_configurable_index_index>
        <reference name="content">
            <block
                type="configurable/configurable"
                name="configurable" template="Svetlana/configurable.phtml" />
        </reference>
    </Svetlana_configurable_index_index>
</layout>
```

- Створимо файл шаблону «template/Svetlana/configurable.phtml», в якому виведемо невпорядкованим списком посилання на отримані товари разом з їх цінами.

```
<?php
$products = $this->getConfigurableProducts();
$helper = Mage::helper('demo');
?>

<h3>Products</h3>

<?php if($products->count() > 0): ?>
    <ul>
        <?php foreach($products as $product): ?>
            <li><a href="<?php echo $product->getProductUrl(); ?>"><?php
echo $product->getData('name'); ?></a>
            <span><?php echo $helper->beautifyPrice($product->getPrice());
?></span></li>
        <?php endforeach; ?>
    </ul>
<?php else: ?>
    <p>No products are available!</p>
<?php endif; ?>
```

Тепер, перейшовши на сторінку «[адреса сайту]/configurable», отримаємо список усіх товарів що конфігуруються (рис.3.8).

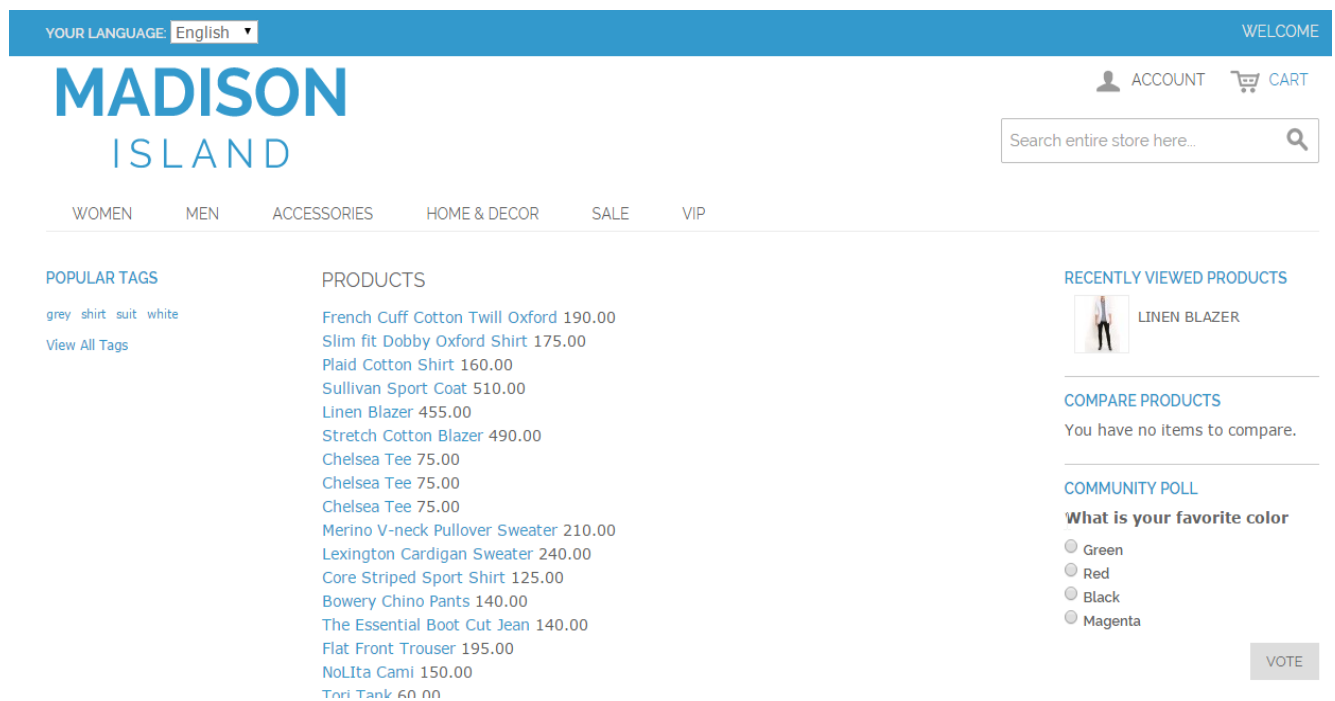


Рис.3.8. Список товарів що конфігуруються

3.9. Створення тем

Багато тем Magento, навіть комерційні, створюються шляхом модифікації файлів чи перевизначенням блоків, які визначені в базовому пакеті. Також можливо створити нову тему з нуля, не використовуючи структурні та контентні блоки, які поставляються в комплекті з Magento. При цьому не обов'язково використовувати header, footer, left, right та content в якості структурних контейнерів. Звісно, створення теми з нуля складніше та займе більше часу, але в залежності від проекту чи інших потреб, це може бути правильним підходом.

Перед початком створення розмітки для магазину, потрібно визначитись із структурним типом сторінок, наприклад:

- Домашня сторінка матиме структуру в три колонки.
- Сторінка зі списком категорій – дві колонки з правою колонкою.
- Сторінки користувачів – дві колонки з лівою колонкою.

Після цього можна створювати розмітку для кожного структурного типу та зберігати її скелетні шаблони в `app/design/frontend/design_package/theme_variation/template/page/`.

Скелетний шаблон названий так через його призначення, бо все що він містить (окрім елементів `<head>`) – це презентаційна розмітка, яка слугує для позиціонування структурних блоків у відповідних областях.

```
<html>
<head></head>
<body>
<div class="header"><?=$this->getChildHtml('header') ?></div>
<div class="middle">
    <div class="col-left"><?=$this->getChildHtml('left') ?></div>
    <div class="col-main"><?=$this->getChildHtml('content') ?></div>
</div>
<div class="footer"><?=$this->getChildHtml('footer') ?></div>
</body>
</html>
```

Рис.3.9. Приклад скелетного файлу теми

Всередині презентаційної розмітки можна помітити виклик РНР методу «`<?=$this->getChildHtml() ?>`» (див. рис.3.9). Таким чином Magento завантажує структурні блоки в скелетні шаблони. Такі виклики здійснюються за ім'ям відповідних структурних блоків. Скелетні шаблони призначаються до магазину за допомогою розкладки.

Після створення скелетних шаблонів, потрібно створити шаблон для кожного контентного блоку.

Magento любить змістовні шаблони. Потрібно розбити розмітку сторінки відповідно до функціоналу сторінки. Наприклад, якщо в дизайні передбачений міні-кошик, розмітку цієї області потрібно помістити в окремий файл шаблону.

Тепер можна перейти до створення розкладки.

Буває два типи розкладок – за замовчуванням та оновлення. Розкладка за замовчуванням (`page.xml`) звичайно використовується на більшості сторінок магазину. Решта файлів розкладок є оновленнями, які просто оновлюють розкладку за замовчуванням для певних сторінок.

Повертаючись до створеного модулю, маємо в якості розкладки за замовчуванням три колонки, які будуть використовуватися на більшості сторінок. Але для сторінки товару нам потрібна інша структура, а саме дві колонки з правою колонкою. Для досягнення цього створимо «catalog.xml», в якому помістимо декілька команд розмітки для завантаження структури з двох колонок замість трьох. Цей процес називається оновленням розкладки.

```
<reference name="root">
  <action method="setTemplate"><template>page/2columns-
right.phtml</template></action >
</reference>
```

Або, скажімо, потрібно за замовчуванням в правій колонці мати блок підписки на розсилку новин, але не на сторінках особистого кабінету користувача. В такому випадку в файлі оновлення розкладки відповідних сторінок (customer.xml) потрібно розмістити команди видалення контентного блоку розсилки.

ВИСНОВКИ

Magento — одна з найпопулярніших відкритих систем для організації електронної комерції в мережі: на базі цієї платформи створено понад 150 тисяч інтернет-магазинів, сторонніми розробниками створено більше 2,5 тисяч розширень, спільнота проекту налічує близько 400 тисяч учасників, код платформи був завантажений більше 3,5 млн разів, через системи на базі Magento кожного року продається товарів більш як на 28 мільярдів доларів. За кілька років свого існування платформа декілька разів була лауреатом премій «Best of Open Source Software Awards» і «SourceForge Community Choice Awards».

Якщо у Вас достатньо простий магазин, кількість товарів у якому не перевищує 100 – 200, то можна використати популярні платформи для інтернет-сайтів WordPress, Joomla або інші, що містять спеціальні додатки для інтернет-магазину та є простими в управлінні.

Однак, якщо необхідно розмістити у магазині досить значну кількість товарів (200–500 і більше), забезпечити зручну навігацію по категоріях і окремих товарах, а також фільтри товарів, то Вам не обійтись без спеціалізованої платформи для інтернет-магазинів – Magento. Перша версія Magento вийшла 2008 році. Нині Magento – найбільш функціональна й ефективна платформа для розробки інтернет-магазинів, що здобула популярність у багатьох країнах світу.

Основними перевагами Magento є:

- можливість задавати різні атрибути (властивості) для кожного товару;
- можливість конфігурувати товар (вибір покупцями кольору, розміру і т.д.);
- існують фільтри і сортування товарів за встановленими атрибутами;
- можливість створення багатьох сайтів на основі однієї системи;
- можливість використання різних валют;
- встановлення податкових ставок;
- багатомовність;
- управління методами оплати і доставки;
- багаторівневе ціноутворення.

Існує велика кількість додатків (модулів) і пакетів дизайну для Magento.

Системам-конкурентам властиві деякі недоліки в порівнянні з Magento, а саме:

- VirtueMart
 - Призначена для веб-сайтів із низьким або середнім рівнем завантаження.
 - У стандартну збірку не входять модулі імпорту/експорту.
 - Багато старого процедурного коду.
- OpenCart
 - Не оптимізована робота SEO.
- Drupal
 - Слабке використання об'єктних можливостей PHP.
- PrestaShop
 - Поступається розміром спільноти та кількістю готових рішень.
- osCommerce
 - Не повністю реалізований механізм шаблонізації.

Одним із недоліків Magento вважають складність освоєння. Технологія призначена для великих проектів, розрахованих на велику кількість користувачів. Створена з використанням Zend Framework, Magento має гнучку архітектуру, модульну структуру та продумані механізми модифікації існуючого функціоналу.

Зручним для роботи команд розробників є розділення представлення та моделей, тобто зовнішнього вигляду та логіки роботи системи.

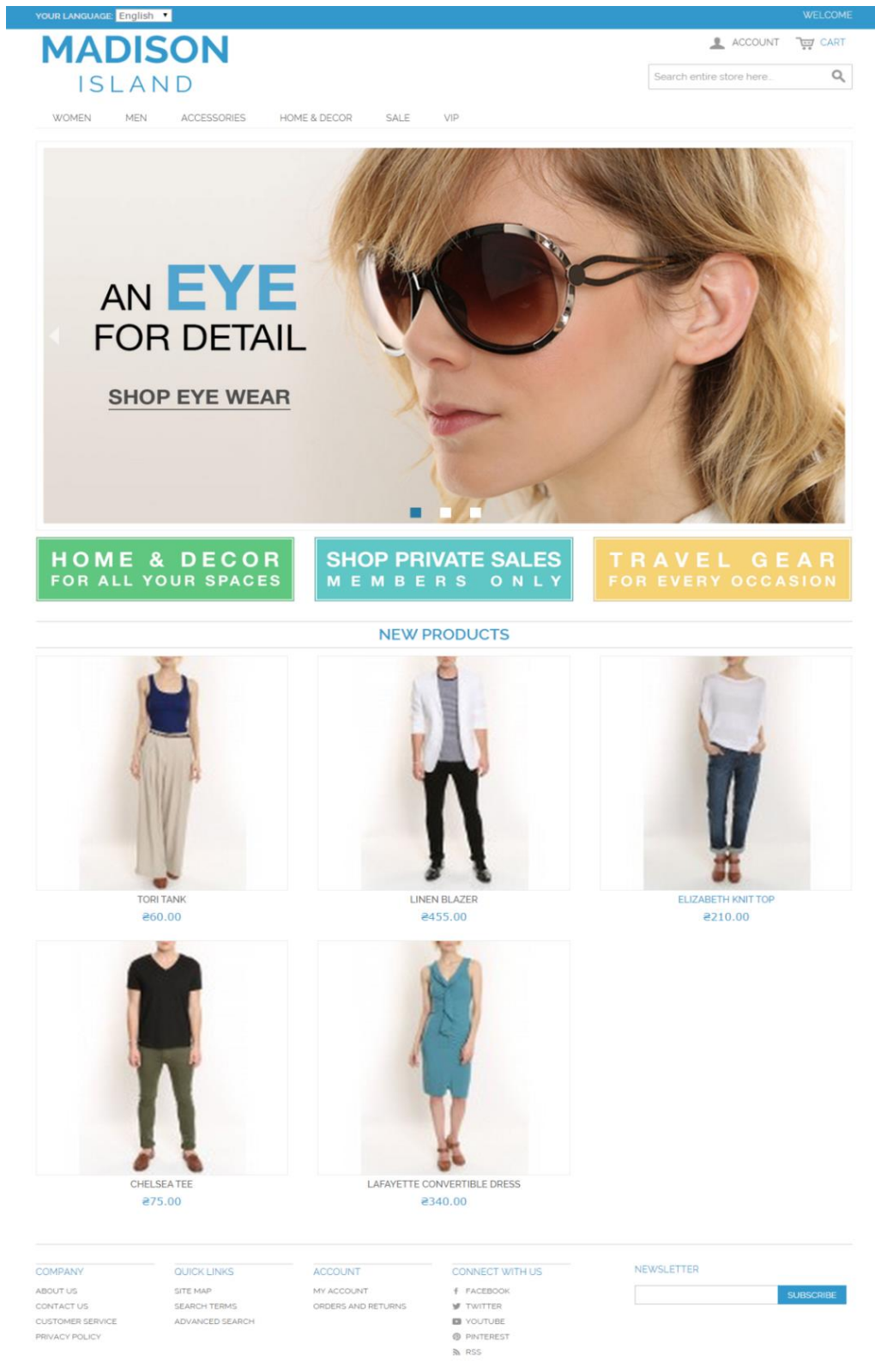
Важливим результатом виконання дипломної роботи є побудова та налагодження технології створення інтернет-магазину з можливостями масштабування асортименту товарів та кількості покупців, причому ця технологія не є надто складною для освоєння та застосування звичайними користувачами.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <http://www.opennet.ru/opennews/art.shtml?num=30814>
2. <http://www.cnet.com/news/and-the-winners-of-the-2018-sourceforge-community-choice-awards-are/>
3. <http://dou.ua/lenta/interviews/interview-magento-2019/>
4. <http://forbes.ua/company/1476>
5. <http://builtwith.com/allo.ua>
6. <http://forbes.ua/magazine/forbes/1336514-v-nachale-bolshogo-vzryva-top-15-internet-kompanij>
7. <http://builtwith.com/mo.ua>
8. <http://wikipedia.org>
9. <http://www.magentocommerce.com/wiki/>
10. Magento Inc. Designer's Guide to "Magento" — 2018.
11. *Milan Stojanov*. Magento Fundamentals. — 2014.
12. *Erich Gamma*, *Richard Helm*, *Ralph Johnson*, *John Vlissides*. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб.: Питер, 2012. — 368 с.
13. Linux-Kernel Archive: Re: Alan Cox quote? (<http://www.uwsg.indiana.edu/hypermail/linux/kernel/0106.2/0405.html>)
14. *Дейт К. Дж.* Введение в системы баз данных — 8-е изд. — М.: Вильямс, 2006. — 1328 с.
15. *Райчев І.Е.* Принципи проектування відкритих розподілених систем / І.Е.Райчев, О.Г.Харченко, В.В.Замковий // Навч. посіб. — К.: Вид-во Нац. авіац. ун-ту "НАУ-друк", 2010. — 240 с.
16. *М. Фаулер*, *К. Скотт*. UML. Основы.
17. *Г. Буч*, *Д. Рамбо*, *А Джекобсон*. Язык UML: Руководство пользователя.
18. *Іванкевич О.В.* Інформаційні системи та структури даних / О.В. Іванкевич, Г.М. Кременецький, В.І. Мазур // Навч.посібник. — К.: НАУ, 2006. — 232с.

ДОДАТОК А

Знімки екрану веб-сайту інтернет-магазину



Продовження Додатку А

YOUR LANGUAGE English

WELCOME

MADISON
ISLAND

ACCOUNT CART

Search entire store here...

WOMEN

MEN

ACCESSORIES


HOME & DECOR

SALE

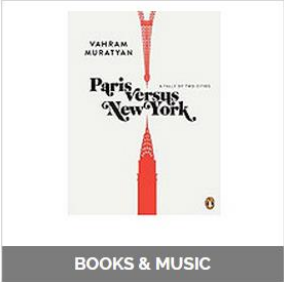
VIP

HOME / HOME & DECOR


HOME & DECOR




HOME
CHIC
HOME
...
*luxury isn't only
for hotel living*




BOOKS & MUSIC



BED & BATH



ELECTRONICS



DECORATIVE ACCENTS

COMPANY

ABOUT US

CONTACT US

CUSTOMER SERVICE

PRIVACY POLICY

QUICK LINKS

SITE MAP

SEARCH TERMS

ADVANCED SEARCH

ACCOUNT

MY ACCOUNT

ORDERS AND RETURNS

CONNECT WITH US

f FACEBOOK

Twitter TWITTER

YOUTUBE

PINTEREST

RSS

NEWSLETTER

SUBSCRIBE

© 2014 Madison Island. All Rights Reserved.

Продовження Додатку А

YOUR LANGUAGE English

WELCOME

MADISON
ISLAND

ACCOUNT CART

Search entire store here...

WOMEN MEN ACCESSORIES HOME & DECOR SALE VIP

HOME / HOME & DECOR / ELECTRONICS

SHOP BY

PRICE

€0.00 - €99.99 (4)

€100.00 - €199.99 (5)

€400.00 - €499.99 (2)

€700.00 and above (1)

COLOR

Black (9)

Charcoal (2)

ELECTRONIC TYPE

Accessories (4)

Camera (3)

Media Players (2)

Speakers + Earphones (3)

POPULAR TAGS

grey shirt suit white

View All Tags

ELECTRONICS

SORT BY: Position VIEW AS: 12 Item(s) SHOW: 12

MADISON LX2200

€400.00

Click for price

★★★★☆

ADD TO CART

Add to Wishlist

Add to Compare

MADISON RX3400

€815.00

Click for price

ADD TO CART

Add to Wishlist

Add to Compare

MP3 PLAYER WITH AUDIO

From: €185.00 To: €275.00

★★★★☆

VIEW DETAILS

Add to Wishlist

Add to Compare

MADISON 8GB DIGITAL MEDIA PLAYER

€150.00

ADD TO CART

Add to Wishlist

Add to Compare

MADISON OVEREAR HEADPHONES

€125.00

★★★★☆

ADD TO CART

Add to Wishlist

Add to Compare

MADISON EARBUDS

€35.00

ADD TO CART

Add to Wishlist

Add to Compare

LARGE CAMERA BAG

€120.00

ADD TO CART

Add to Wishlist

Add to Compare

16GB MEMORY CARD

€30.00

ADD TO CART

Add to Wishlist

Add to Compare

8GB MEMORY CARD

€20.00

ADD TO CART

Add to Wishlist

Add to Compare

CAMERA TRAVEL SET

From: €445.00 To: €965.00

VIEW DETAILS

Add to Wishlist

Add to Compare

3-YEAR WARRANTY

€75.00

ADD TO CART

Add to Wishlist

Add to Compare

5-YEAR WARRANTY

€100.00

ADD TO CART

Add to Wishlist

Add to Compare

SORT BY: Position VIEW AS: 12 Item(s) SHOW: 12

RECENTLY VIEWED PRODUCTS

TITIAN RAW SILK PILLOW

COMPARE PRODUCTS

You have no items to compare.

COMMUNITY POLL

What is your favorite color

Green

Red

Black

Magenta

VOTE

COMPANY

ABOUT US

CONTACT US

CUSTOMER SERVICE

PRIVACY POLICY

QUICK LINKS

SITE MAP

SEARCH TERMS

ADVANCED SEARCH

ACCOUNT

MY ACCOUNT

ORDERS AND RETURNS

CONNECT WITH US

FACEBOOK

TWITTER

YOUTUBE

PINTEREST

RSS

NEWSLETTER

SUBSCRIBE

© 2014 Madison Island. All Rights Reserved.

Продовження Додатку А

YOUR LANGUAGE: English

WELCOME


MADISON
ISLAND

ACCOUNT CART (1)

Search entire store here.

WOMEN MEN ACCESSORIES HOME & DECOR SALE VIP

HOME / HOME & DECOR / ELECTRONICS / MP3 PLAYER WITH AUDIO



MP3 PLAYER WITH AUDIO

From ₺185.00
To ₺275.00
Price as configured: ₺150.00

★★★★☆
3 Review(s) | Add Your Review

IN STOCK

Pick up your Media Player and Audio Output together.

Media Player *
Madison 8GB Digital Media Player +₺150.00

Qty: 1

Audio Output *
Choose a selection...

Qty: 0

Qty: 1

ADD TO CART

Add to Wishlist


Add to Compare

✉

f

🐦

MORE VIEWS



DESCRIPTION REVIEWS

CUSTOMER REVIEWS 3 ITEM(S)

LIKED IT, WOULD LIKE TO HAVE CHOSEN COLOR

I wasn't able to choose the color but other than that a good experience and very easy

PRICE ★★★★★
VALUE ★★★★★
QUALITY ★★★★★

REVIEW BY JOHN / (POSTED ON 5/28/2013)

GOOD BATTERY LIFE

Battery life is good

PRICE ★★★★★
VALUE ★★★★★
QUALITY ★★★★★

REVIEW BY JOHN / (POSTED ON 5/28/2013)

THANK YOU!

I just love this little device and it holds plenty of music I can travel with and share with friends. Thank you for such wonderful pricing.

PRICE ★★★★★
VALUE ★★★★★
QUALITY ★★★★★

REVIEW BY GIL H / (POSTED ON 5/28/2013)

PRODUCT TAGS

Add Your Tags:

ADD TAGS

Use spaces to separate tags. Use single quotes (') for phrases.

COMPANY

ABOUT US
CONTACT US
CUSTOMER SERVICE
PRIVACY POLICY

QUICK LINKS

SITE MAP
SEARCH TERMS
ADVANCED SEARCH

ACCOUNT

MY ACCOUNT
ORDERS AND RETURNS

CONNECT WITH US

f FACEBOOK
🐦 TWITTER
📺 YOUTUBE
📌 PINTEREST
📡 RSS

NEWSLETTER

SUBSCRIBE

© 2014 Madison Island. All Rights Reserved

Продовження Додатку А

YOUR LANGUAGE: English

WELCOME

MADISON
ISLAND

[WOMEN](#) [MEN](#) [ACCESSORIES](#) [HOME & DECOR](#) [SALE](#) [VIP](#)





[ACCOUNT](#) [CART \(2\)](#)

Search entire store here...

SHOPPING CART

PROCEED TO CHECKOUT

MP3 Player with Audio was added to your shopping cart.

PRODUCT	PRICE	QTY	SUBTOTAL	
 <div>MADISON RX3400 SKU: hde003</div>	€715.00	<input type="text" value="1"/> Edit	€715.00	
 <div>MP3 PLAYER WITH AUDIO SKU: hde014-hde012-hde011 <i>Media Player:</i> 1 x Madison 8GB Digital Media Player €150.00 <i>Audio Output:</i> 1 x Madison Overear Headphones €125.00</div>	€275.00	<input type="text" value="1"/> Edit	€275.00	

[EMPTY CART](#) [UPDATE SHOPPING CART](#) -OR- [CONTINUE SHOPPING](#)

DISCOUNT CODES [APPLY](#)

ESTIMATE SHIPPING AND TAX

COUNTRY *
United States

STATE/PROVINCE *
Please select region, s

CITY
ADDRESS_HOME_CIT

ZIP *
ADDRES

[ESTIMATE](#)

SUBTOTAL €990.00
TAX €81.68

GRAND TOTAL €1,071.68

PROCEED TO CHECKOUT

COMPANY

[ABOUT US](#)
[CONTACT US](#)
[CUSTOMER SERVICE](#)
[PRIVACY POLICY](#)

QUICK LINKS

[SITE MAP](#)
[SEARCH TERMS](#)
[ADVANCED SEARCH](#)

ACCOUNT

[MY ACCOUNT](#)
[ORDERS AND RETURNS](#)

CONNECT WITH US

[FACEBOOK](#)
[TWITTER](#)
[YOUTUBE](#)
[PINTEREST](#)
[RSS](#)

NEWSLETTER

[SUBSCRIBE](#)

© 2014 Madison Island. All Rights Reserved.

Продовження Додатку А

YOUR LANGUAGE: English

WELCOME

MADISON
ISLAND

ACCOUNTCART (2)

Search entire store here...

WOMENMENACCESSORIESHOME & DECORSALEVIP

CHECKOUT

1CHECKOUT METHODEdit

2BILLING INFORMATION

First Name *Required Fields

NAME_FIRST

Last Name *

NAME_LAST

Company

COMPANY_NAME

Email Address *

EMAIL_ADDRESS

Address *

ADDRESS_HOME_LINE1

Street Address 2

ADDRESS_HOME_LINE2

City *

ADDRESS_HOME_CITY

State/Province *

Please select region, state or province

Zip/Postal Code *

ADDRESS_HOME_ZIP

Country *

United States

Telephone *

PHONE_HOME_WHOLE_NUMBER

Fax

UNKNOWN_TYPE

☒ Ship to this address

☐ Ship to different address

CONTINUE

3SHIPPING INFORMATION

4SHIPPING METHOD

5PAYMENT INFORMATION

6ORDER REVIEW

YOUR CHECKOUT PROGRESS

BILLING ADDRESS

SHIPPING ADDRESS

SHIPPING METHOD

PAYMENT METHOD

COMPANY

ABOUT US

CONTACT US

CUSTOMER SERVICE

PRIVACY POLICY

QUICK LINKS

SITE MAP

SEARCH TERMS

ADVANCED SEARCH

ACCOUNT

MY ACCOUNT

ORDERS AND RETURNS

CONNECT WITH US

f FACEBOOK

TWITTER

YOUTUBE

PINTEREST

RSS

NEWSLETTER

SUBSCRIBE

© 2014 Madison Island. All Rights Reserved.

Закінчення Додатку А

This is a demo store. Any orders placed through this store will not be honored or fulfilled.

Global Record Search

Logged in as admin | Monday, February 9, 2015 | [Log Out](#)

Dashboard

Sales

Catalog

Customers

Promotions

Newsletter

CMS

Reports

System

Get help for this page

Orders

Create New Order

Page 1 of 3 pages | View 20 per page | Total 46 records found | [New Order RSS](#)

Export to: CSV | Export | Reset Filter | Search

Select Visible | Unselect Visible | 0 items selected

Actions

Submit

	Order #	Purchased From (Store)	Purchased On	Bill to Name	Ship to Name	G.T. (Base)	G.T. (Purchased)	Status	Action
Any			From: To:			From: To:	From: To:		
<input type="checkbox"/>	14500004-1	Main Website Madison Island English	May 4, 2014 3:36:47 AM	Jane Doe	Jane Doe	\$975.55	\$975.55	Canceled	View
<input type="checkbox"/>	145000004	Main Website Madison Island English	May 4, 2014 3:19:24 AM	Jane Doe	Jane Doe	\$975.55	\$975.55	Canceled	View
<input type="checkbox"/>	145000003	Main Website Madison Island English	Apr 30, 2014 11:42:28 AM	Jay Smith	Jay Smith	\$372.38	\$372.38	Pending	View
<input type="checkbox"/>	145000002	Main Website Madison Island English	Apr 30, 2014 11:40:36 AM	Jay Smith	Jay Smith	\$372.38	\$372.38	Canceled	View
<input type="checkbox"/>	100000211	Main Website Madison Island English	May 30, 2013 8:55:38 AM	John Doe	John Doe	\$595.45	\$595.45	Pending	View
<input type="checkbox"/>	100000210	Main Website Madison Island English	May 29, 2013 7:47:04 PM	John Doe		\$7.45	\$7.45	Complete	View
<input type="checkbox"/>	100000208	Main Website Madison Island English	May 29, 2013 7:29:44 PM	John Doe	John Doe	\$741.20	\$741.20	Processing	View
<input type="checkbox"/>	100000206	Main Website Madison Island English	May 29, 2013 7:28:25 AM	John Doe	John Doe	\$1,247.64	\$1,247.64	Complete	View
<input type="checkbox"/>	100000205	Main Website Madison Island English	May 28, 2013 6:10:20 AM	Jane Smith	Jane Smith	\$447.14	\$447.14	Complete	View
<input type="checkbox"/>	100000204	Main Website Madison Island English	May 28, 2013 6:08:58 AM	Jane Smith	Jane Smith	\$333.21	\$333.21	Processing	View
<input type="checkbox"/>	100000203	Main Website Madison Island English	May 28, 2013 6:07:47 AM	Jane Smith	Jane Smith	\$432.26	\$432.26	Complete	View
<input type="checkbox"/>	100000202	Main Website Madison Island English	May 26, 2013 3:43:25 AM	Jane Doe	Jane Doe	\$950.20	\$950.20	Pending	View
<input type="checkbox"/>	100000201	Main Website Madison Island English	May 26, 2013 3:37:05 AM	Jane Doe	Jane Doe	\$387.27	\$387.27	Pending	View
<input type="checkbox"/>	100000200	Main Website Madison Island English	May 25, 2013 7:58:37 AM	John Doe	John Doe	\$599.25	\$599.25	Pending	View
<input type="checkbox"/>	100000199	Main Website Madison Island English	May 25, 2013 7:42:11 AM	Jane Doe	Jane Doe	\$453.20	\$453.20	Complete	View
<input type="checkbox"/>	100000135	Main Website Madison Island English	Apr 25, 2013 11:41:25 PM	Dong Berg	Dong Berg	\$14,493.00	\$14,493.00	Complete	View
<input type="checkbox"/>	100000111	Main Website Madison Island English	Apr 25, 2013 4:10:44 AM	Kip Chee	Kip Chee	\$3,499.17	\$3,499.17	Processing	View
<input type="checkbox"/>	400000002	Private Sales Website Private Sales Store Private Sales View [deleted]	Apr 24, 2013 8:32:54 PM	Pearl Glory	Pearl Glory	\$1,566.11	\$1,566.11	Processing	View
<input type="checkbox"/>	400000001	Private Sales Website Private Sales Store Private Sales View [deleted]	Apr 24, 2013 8:21:49 PM	Pearl Glory	Pearl Glory	\$724.62	\$724.62	On Hold	View
<input type="checkbox"/>	100000095	Main Website Madison Island English	Apr 24, 2013 10:12:51 AM	Gogi Cooper	Gogi Cooper	\$3,795.31	\$3,795.31	Pending	View

Help Us Keep Magento Healthy - Report All Bugs

Interface Locale: English (United States) / English

Magento ver. 1.9.1.0

Connect with the Magento Community

Magento™ is a trademark of Magento Inc.

Copyright © 2015 Magento Inc.